khazanah informatika
Jurnal Ilmu Komputer dan Informatika

Rice Seedling Image Classification Using Light Convolutional Neural Network

Indra Hermawan [1], Maria Agustin [2]*, Ayu Rosyida Zain [3], Meutia Tri Mulyani [4], David Nathanael [5]
*maria.agustin@tik.pnj.ac.id
[1,2,3,4,5]Department of Informatics and Computer Engineering,
Politeknik Negeri Jakarta
Depok

Abstract-The need for food, especially rice, continues to increase. Therefore, a production increase of around 70% is required to meet the demand. In this case, supervision and care of rice from planting must be carried out efficiently, which can be done by deep learning, namely Convolutional Neural Networks (CNN). The classification was carried out on the image of rice seedlings in the form of rice seedlings and bare land patch images. The main purpose of this research is to conduct a comparison test of the performance of each CNN model with a lightweight architecture and validate the architecture. A lightweight CNN architecture is used due to its lower architecture size but still has decent performance compared to the regular CNN model for the rice seedlings dataset. Training and testing were carried out on the Rice Seedling Dataset to determine the performance of the proposed method. The research was built using the PyTorch library and the Python programming language and resulted in 99% of accuracy, precision, recall, kappa, and F-1 Score. In addition, validation was carried out using K-Fold Cross Validation which also had the best accuracy of 99%. Therefore, we conclude that the developed model can properly classify images of rice seedlings and arable land.

Keywords: lightweight convolution neural network; rice seedling; arable land; image classification; farming

Article info: *submitted November, 10, 2022, revised June, 1, 2023, accepted August, 16, 2023*

1. Introduction

The global increase in the world's population has led to an increasing need for rice as the main daily food [1]. Production of agricultural products is also required to increase by around 70% of the predicted number of demands. One example of the most important food in the world is rice. Rice is very important as people's main food in Asia, where rice is ranked second as a daily staple food [2]. The growing population has resulted in soaring food production to meet the ever-increasing demand, which becomes a challenging task. The importance of good quality rice in considerably sufficient amounts requires monitoring for the growth of the rice plants, especially since the rice seedling period. Therefore, estimating rice growing conditions and evaluating the yield is essential in rice farming [3].

Researchers have developed several smart solutions to deal with the said problem, such as better farming techniques, precision farming, agricultural automation, and so on [1]. One of the most stands out implementations carried out by researchers to overcome these problems is artificial intelligence. Currently, the artificial intelligence revolution has raised the standard of living in every sector, including the agricultural sector. One example of artificial intelligence to implement is deep learning [4]. Previous studies have implemented deep learning on rice growth analysis, mainly focused on rice seedlings, which is further discussed in Table 1.

The first paper is a research on creating a dataset of rice seedlings for deep learning practice, which is also used as a dataset in our research, accompanied by a showcase of implementation using transfer learning, namely VGG-16 [5]. The next research is a study that focused on the difficulty in counting rice seedlings in a paddy field, which then inspired the writer to build a lighter YOLOv4 object detection model, namely YOLOv4-L1 [3]. Another research by Ma, Xu et al. studied the application of rice images on the semantic segmentation model based on FCN [6]. In the next rice seedlings research, multiple transfer learning methods were used on rice images using EfficientDet-D0 and Faster R-CNN, which is compared with the HOG-SVM model [7]. Finally, the last research focused on aerial imagery of rice images used transfer learning from two types of machine learning, namely EfficientDet-D0 and Faster R-CNN [8].

Table 1. List of research about rice seedlings image processing

Column title	Problems	Method	Result
Yang, M., 2021 [5]	Creating labeled and unlabeled data to provide a UAV image dataset of rice paddy by findable and accessible through domain-specific repositories with a showcase of CNN classification.	Using modified VGG-16, which is a classical CNN architecture to train the patch-based rice seedling classification dataset for the CNN showcase.	An accuracy of 0.99 was achieved by all divisions of the cross-validation dataset. The rice seedling dataset provides the training and validation dataset, patch-based detection samples, and the ortho mosaic image of the field.
Li, H., 2022 [3]	The difficulty of counting rice seedlings, considering how many other objects such as weeds exist within the images might cause counting errors in traditional image processing methods.	Proposed modified YOLOv4 model, namely YOLOv4-L1. The model is adjusted to be lightweight to reduce the training process and consumed time.	By comparing to the original YOLOv4 model, the YOLOv4-L1 model managed to reduce 2.45 hours of training time using similar counting results.
Ma, X., 2019 [6]	Detecting site-specific rice seedlings and weeds to reduce the over-application of herbicide in paddy fields.	Proposed semantic segmentation method with the SegNet which is based on a fully convolutional network (FCN).	SegNet method achieved an average accuracy rate of 92.7%, while the average accuracy rates of FCN and U-Net methods were respectively 89.5% and 70.8%.
Tseng, H., Yang, M., 2022 [7]	Crop detection based on computer vision with UAV in precision agriculture.	Transfer learning EfficientDet-D0 and Faster R-CNN is applied. The result is compared to histograms of oriented gradients (HOG)-based support vector machine (SVM) classification.	The CNN-based models perform better compared to the HOG-SVM model, with 10% higher mAP and mIoU. Additionally, the modelcomputation also outperformed HOG-SVM with 1000 times faster computation speed.
Anuar, M. M., Halin, A. A., 2022 [8]	Optimal planting density analysis for paddy cultivation by detecting defective paddy rice seedlings to alleviate the burden of traditional and laborious methods.	Several deep convolutional neural networks (DCNN) models were explored to determine which model performs the best for defective paddy seedlings detection with aerial imagery.	By using one-stage pre-trained object detectors EfficientDet-D1 and EficientNet,the proposed methods show that defective paddy rice seedlings were successfully detected with the highest precision and F1-Score respectively 0.83 and 0.77.

Some of the studies previously mentioned the use of CNNs, which applied transfer learning such as VGG-16 and EfficientDet-D0. The use of CNN models such as transfer learning can achieve very good performance, but this achievement requires a lot of trade-offs in the architectural model. In this case, modifications such as the constant addition of a network layer are unavoidable with the increasing demands for better performance. This can cause problems in several aspects, including storage space and speed problems, especially in future practical applications such as embedded systems [9].

To overcome the problems mentioned previously, in this research, a lightweight convolutional neural network (LCNN) is proposed, which is a CNN model with limited weight and storage, but can achieve performance that can compete with more complex models [10]. In addition, there are several advantages of using LCNN including (1) requiring less server-to-server communication in distributed training, (2) exporting new model from the cloud to an autonomous vehicle took less bandwidth, (3) Field-Programmable Gate Array (FPGAs) and hardwares with limited memory can deploy smaller CNN better [10].

Table 2. List of research about LCNN

Column title	Problems	Method	Result
Polsinelli, M., 2020 [11]	An automated method for recognizing images of the lungs of people with COVID-19 and without COVID-19.	Using LCNN based on the SqueezeNet model of COVID-19 CT images based on pneumonia and healthy CT images by dividing the training into 4 experiments with different models.	CNN-2 achieves 85.03% accuracy, 87.55% sensitivity, 81.95% specificity, 85.01% precision, and 86.20% F1-Score, where CNN-2 has good efficiency compared to other CNNs.
Yang, L., 2019 [12]	Hand detection supports many activities, including HCI applications which can solve problems such as background, motion blur, and lighting changes.	LCNN is proposed, which uses a modified MobileNet as a feature extractor along with an SSD framework for fast and powerful hand position and orientation detection.	This method achieves an average precision of 83.2% with 139 FPS on the NVIDIA Titan X.
Kang, Q., Zhao, 2020 [10]	Use thermal infrared cameras in the Driver Assistance System (DAS) field and discuss their performance in vehicle recognition at night.	CNN Net1 is designed with four convolution layers to classify 4 vehicle classes. Fire module as the core structure of SqueezeNet, was also implemented to build nine LCNNs (Net2–Net10) to compare with Net1.	It was concluded that Net9 was the most optimal network with a classification accuracy of 97% with 10.6% of the total parameters of Net1. Furthermore, one image recognition time is only 0.52 ms, so it can be applied to embedded systems such as DAS.

There are several studies related to LCNN that will be elaborated in Table 2. The first related research was designed by Polsinellia, Matteo et al., namely the method of recognizing images of the lungs of patients with COVID-19 using LCNN with the SqueezeNet model [11]. Furthermore, Yang, Li et al. use LCNN in hand detection [12]. Finally, a study by Kang, Qing, et al. about detecting vehicles on thermal infrared images using LCNN [10].The objective of this study is to build an image classification architecture with an LCNN model for rice seedlings images. The proposed research will use secondary datasets sourced from GitHub with classes including arable land and rice seedlings. Previously, related publications have been published regarding similar datasets on [5], where in this study a convolutional neural network (CNN) was used. The CNN model itself is a modification of the VGG-16 algorithm architecture with the overall performance of the training result is 0.99. Based on the publication, this research will focus on making improvements to machine learning that was previously built to classify rice seedlings using the Rice Seedlings dataset by designing a more streamlined and concise architecture supported by theoretical analysis, yet still maintaining the performance from the previously conducted research.

The novelty and contribution that will be described in this paper are the construction of the LCNN model which is implemented on patch-based rice seedlings and bare land images. The main purpose of this research is to conduct a comparison test of the performance of each CNN model with a lightweight architecture and validate the architecture using K-Fold Cross Validation.

We will divide this research into several chapters Chapter 1 as the introduction. Chapter 2 explains the research method, containing the dataset, type of classification used, and performance evaluation. Finally, the results and discussion in Chapter 3 describe the analysis of the training on the architectural model that has been developed and ends with conclusions.

2. Methods

a. Dataset

The datasets used are released on GitHub and are available in [13] (last accessed 10 October 2022). The classes on datasets were divided in self-classification into two labels, namely rice seedling and arable land as shown in Figure 1, Each class has approximately 28 K and 26.5 K samples. The total sample of the dataset is about 54.6 K, with the size of each image 48 x 48 pixels and has three types of visible bands, namely R, G, and B [5].

Figure 1. Sample of two labels on a dataset

Each class is then divided into a training dataset, validation dataset, and testing dataset. In full, the training data has around 43703 sample images, validation data of about 1093 sample images, and testing data of 9832 sample images. Table 3 displays the sample size of each class, specifically from rice seedlings, arabe land, and the total number.

Table 3. Table of Datasets

Classes	Training Dataset	Validation Dataset	Testing Dataset	Total Dataset
Arable Land (bare land)	22,438	561	5,048	28,047
Rice Seedlings	21,265	532	4,784	26,581
Total	43,703	1,093	9,832	54,628

b. Convolutional Neural Network

Deep Learning is an important branch of Machine Learning where an artificial neural network is built to learn algorithms inspired by the workings of the human brain from a large amount of data [14]. In image classification and object detection, Deep Learning has many advantages over traditional Machine Learning [15]. The Deep Learning technique can extract informative features in the original data using hierarchical layers. The learning process runs automatically so that Deep Learning can be implemented in various situations [16].

The popularity of deep learning is growing very fast in big data analytics. Various discoveries and breakthroughs emerged with the implementation of deep learning in various fields of computer vision, including image classification, object detection, and natural language processing [16]. Deep learning techniques also produce many benefits in the agricultural sector. Previous studies by Ming et al. developed deep learning for open rice datasets. It is past research that we will further improve on existing performance with the results of the Deep Learning trials that will be carried out. This study uses CNN architecture to detect patch-based rice seedlings in the Rice Seedlings image dataset from GitHub [5].

Using Deep Learning has more advantages than approaches with Machine Learning for methods of image classification, object detection, and localization. This is because Deep Learning can learn powerful features. This method is able to effectively extract features from the image to be trained, which in this case are arable land and rice seedlings [15].

Deep Learning has several available architectures, but the common one is the Convolutional Neural Network (CNN) which is currently popular and often used [17]. The typical use of CNN is to analyze and classify images. The word convolution refers to the filtering process. Stacks or layers of convolutional layers are the basis of CNN. Each layer receives input data, transforms, or convolutes it, and produces an output which is then sent to the next layer. This convolution operation will simplify the data to be processed and realized better.

c. Rice Image Classification

Image classification is performed on the dataset using the CNN algorithm. A fairly simple network structure minimizes the number of convolution layers, filters, and fully connected layers, which reduces the number of parameters in the training phase. The structure is illustrated based on the previous research related to LCNN as shown in Figure 2 [18].

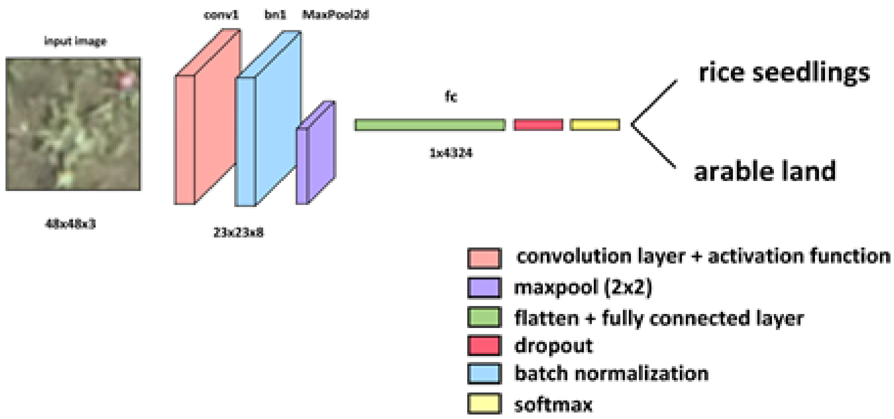

Figure 2. Proposed LCNN Architecture

Table 4 describes each layer and its parameters in the built network model. In the initial convolution layer, both consist of 8 filters and a kernel size of 3x 3 pixels. For the next convolution layer experiment, the filters were increased by 16 and 32. Each convolution layer is followed by an activation function operation, rectified linear unit (ReLU), or Sigmoid. In addition, the convolution operation also uses same-padding, where this method enlarges the pixel boundaries before the convolution operation to maintain the size with the input tensor. The next convolution layer is followed by a batch-normalization operation and a max pooling layer with a 2x2 pixel kernel. Then before entering the fully connected layer, the flatten function is performed.

Table 4. Table of the proposed architecture

Layer	Parameter	Activation Function
Input	48×48×3	-
Convolutional Layer	8 filter (3×3), 1 stride, same padding	Yes
Batch normalization 1 (Bn1)	-	-
Pooling 1 (Pool1)	Max pooling (2×2)	-
Flatten	-	-
Fully connected layer (fc)	128, 64, 32 nodes	Yes
Dropout	-	-
Output	2 nodes	Softmax

The first fully connected layer consists of predetermined nodes, namely 128, and subsequent nodes 64 and 32. Like the convolution layer, the fully connected layer is followed by an activation function (ReLU or Sigmoid) and dropout operations. We used dropout to eliminate overfitting, as this operation will result in only a few random trained active neurons. We configure dropout at 0.1 rates. In the final stage, we will represent two classes of images, on the rice seedling dataset. The output layer of the architecture is a softmax activation function with an output value equal to 1.0. This activation function also limits each output value to be between 0-1, which in other words, is the probability of each class. We conducted the training process in 10 epochs, and the architecture was built using PyTorch. The training was done multiple times by changing the architectural model parameters, where the parameter values that became the reference can be seen in Table 5. We conducted the research by changing the value of the parameters in each training. Then we stored the training process results to be analyzed and compared to determine which architecture performs best.

Table 5. Table of tested parameters

Parameter	Value
Convolutional Layer	8, 16, 32 filters
Hidden Layer	128, 64, 32 nodes
Batch Size	32
Dropout	0.1
Activation Function	Sigmoid, ReLU
Optimizer	SGD, Adam
Learning rate	0.01, 0.001, 0.0001

d. Evaluation Parameter

1) Precision
Precision is the correct classification ratio to the total number of classifications. Conversely, low precision indicates a large number of false positives. Therefore, we represent Precision as follows:

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

2) Recall
Recall is the ratio of the number of correct classifications to the number of the sample. A high recall indicates a small number of misclassified samples. We represent Recall as follows:

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

3) Accuracy
Accuracy is a fraction of truth in the model which is calculated as the sum of the correct classifications divided by all the classifications. We represent Accuracy as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

4) F1-Score
The F1-score quantifies the harmonic average between precision and accuracy. This metric usually represents the robustness of the classification and can be calculated using the following formula:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (1)$$

5) Cohen's Kappa
Cohen's Kappa scores are compared to expected values, especially when the two assessments are independent. The numerator represents the gap between the observed probability of success and the probability of success in the case that is assumed to be very bad.
In evaluation, the value of 0 indicates no agreement, 0.01-0.20 as none to little, 0.21-0.40 as fair, 0.41-0.60 as moderate, 0.61-0.80 as substantial, and 0.81-1.00 as almost perfect [19]

e. Device Setup

The research was done using Google Collaboratory or Google Colab, one of Google's services that do not require additional setup and work in the cloud, so computer specifications are not considered as long as Google Chrome is running. With Google Colab, users can write and execute code and save it just like using environments such as Jupyter Notebook and MATLAB [20].

3. Result and Discussion

In this section, the results of the experiments will be elaborated with analysis using tables, graphs, and comparative analysis on the selected models.

a. Training Results Analysis

In this study, the author uses Python (Programming Language) as the main tool to create the LCNN model. Some help libraries include Pytorch, Matplotlib, Torchmetrics, and Numpy. The author uses the method to create the model as an experiment with tuning where training is carried out by gradually changing the existing parameters. In addition, restrictions were placed on the architectural model the author designed, as listed in Table 4 previously, to make it light in terms of computation and structure while still getting good accuracy.

Based on the values determined, the experiment was carried out 72 times using various types of architecture with tuning to several parameters. From the results of the training, the five best results from the previously designed architecture are shown in Table 6. The best models are selected based on the measurement of the final loss from each model at the end of the training process, which must be as low as possible. In addition, it is ensured that between training loss and validation loss, there is no significant difference to prevent overfitting of the model.

Table 6. 5 LCNN architectures with the best loss reduction of the training process

Model No.	Architecture	Activation Function	Optimizer	Learning rate	Training Loss	Test Loss
(1)	1 Convolution 3 Hidden Layer	ReLU	Adam	0.0001	0.000023	0.00041
(2)	2 Convolution 1 Hidden Layer	ReLU	Adam	0.001	0.000087	0.000266
(3)	**2 Convolution 1 Hidden Layer**	**ReLU**	**Adam**	**0.0001**	**0.000019**	**0.000043**
(4)	2 Convolution 2 Hidden Layer	ReLU	SGD	0.01	0.012304	0.037297
(5)	2 Convolution 2 Hidden Layer	ReLU	Adam	0.001	0.001402	0.003633

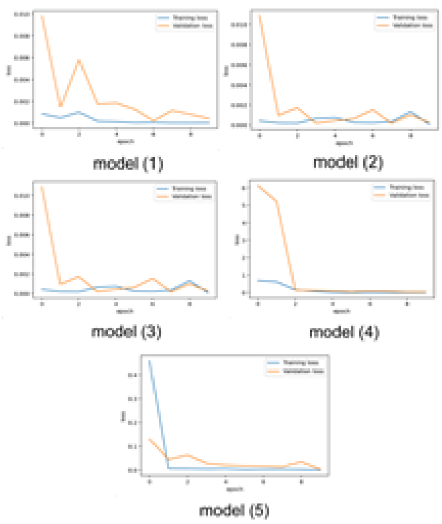

Figure 3. Loss comparison of the five best models

The parameters tuned during the training process as well as the training loss and testing loss are elaborated in Table 6. The five best models have similarities in the activation function choice, where all of them use ReLU. In addition, the majority of the models use Adam as the optimizer. Out of all the best-shown models, the model with the best loss reduction is model (3), where the training loss and testing loss is almost 0, followed by model (1) and model (3) with similar amount of loss and slightly different between losses in the model. The least good result shown is model (4) which has significantly higher losses compared to the others by almost 1000 times. Based on the training data, model (4) has poor performance and when compared to the other models, it used a different optimizer, where SGD was used.

As a comparison of loss reduction during the training process, graphically the training process is also carried out, as in Figure 3. This ensures that each selected architecture does not experience overfitting or underfitting. Each model experienced a steady decrease in loss, especially in model (3), (4), and (5). The loss reduction is steadily going down with only occasional instability of increasing loss. On the other hand, model (1) and (2) experienced a slight instability in the loss during the training process, but not too significant.

Table 7. 5 LCNN architectures with the best performance of the training process

Model No.	Architecture	Precision	Recall	F1-Score	Kappa	Accuracy
(1)	1 Convolution 3 Hidden Layer	0.9999	0.9999	0.9999	0.9998	0.9999
(2)	2 Convolution 1 Hidden Layer	1	0.99	1	0.99	1
(3)	**2 Convolution 1 Hidden Layer**	**1**	**1**	**1**	**1**	**1**
(4)	2 Convolution 2 Hidden Layer	0.8862	0.9019	0.8718	0.8035	0.9017
(5)	2 Convolution 2 Hidden Layer	0.9991	0.9991	0.9991	0.9981	0.9991

In addition to analyzing the loss of the model, the selection of the best model is based on several evaluation parameters which is precision, recall, F1-Score, kappa, and accuracy shown in Table 7. The value of each parameter from the lowest to the highest is 0 to 1. Model (3) is the most prominent one with values for all parameters reaching 1 or reaching the perfect score for every parameter, followed by model (2) with precision, F1-score, and accuracy is 1 and other parameters are 0.99 or close to perfect. Unfortunately, model (4) has decreased in performance compared with the reduced loss, where the evaluation parameters of precision, recall, F1-Score, kappa, and accuracy each cannot reach an average of above 0.95 compared to the other models.

Similar to loss analysis, the increase in performance during the training process is shown in graphical form to facilitate comparison, which is in Figure 4. Models (2) and (3) experienced the best improvement for the five measurements, where the measurements that occurred were stable and continued to increase. Unfortunately, in model (1) and model (4), the decline at one point is quite severe, while for model (5), kappa performance is far behind compared to other measurements.

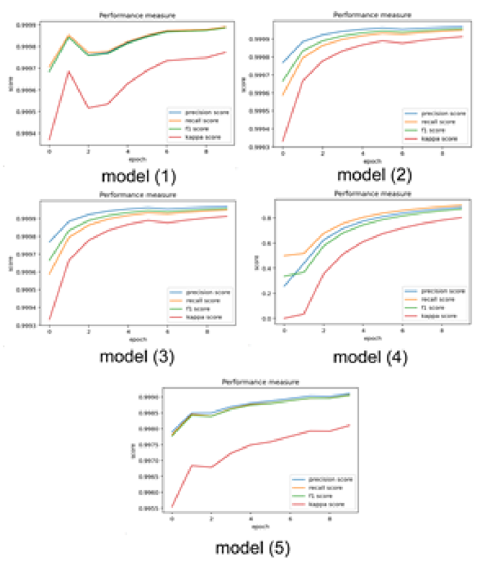

Figure 4. Evaluation parameter result of five best models

Finally, further trials will be carried out with K-Fold Cross Validation to validate and determine the best model out of the five shown results.

b. K-Fold Cross Validation

K-Fold Cross Validation is a method to validate a dataset where it will evaluate and test the performance of the model that has been built [21]. K-Fold Cross Validation works where a dataset will be divided into several K called folds, where each fold is then used as a set to test several points in a computer vision [22]. In this study, K-fold Cross Validation will be used to validate whether the model has good performance in various rice seedlings dataset combinations [23]. This method is carried out with the use of 5-fold. The evaluation results using K-Fold Cross Validation on the five best results are in Table 8.

Table 8. K-Fold Cross Validation Result

Model No.	Fold 1	2	3	4	5	Average
(1)	60%	64%	83%	88%	67%	73%
(2)	99%	99%	48%	99%	47%	79%
(3)	99%	99%	99%	99%	50%	89%
(4)	98%	49%	49%	99%	47%	69%
(5)	99%	99%	100%	99%	99%	99%

Of the 5 selected architectures, one of the five architectures, namely model (4), received the lowest accuracy of only 69% accuracy, followed by model (1) and model (2), with the result value of 73% and 79% respectively. However, the other two models managed to get scores above 80%, so the architecture has good and decent performance from the K-Fold Cross Validation.

Based on the data in Table 8. and by considering the analysis result of the losses obtained at the end of the training process and evaluating the existing performance from the previously shown graph in Figure 4, model (3) is the best in terms of performance and data validation with overall performance reaching almost 100% and K-Fold Cross Validation of 89%. This model produces more stable and high performance compared to other the other K-fold result. Model (5) for example, has the best k-fold result which is at 99%, but the performance result between the parameters that were previously produced in the training process is imbalanced, especially in the kappa performance, which is why model (3) is better and chosen as the best-trained model out of the five chosen LCNN architectures.

4. Conclusion

The trial of the development of LCNN was implemented on a rice seedlings dataset, resulting in a binary image classification between bare land and rice seedlings with a lightweight architecture with very good performance. This paper contributes to the construction of the LCNN model which is able to be implemented on patch-based rice seedlings and bare land images. By testing the LCNN model with different parameters, the evaluation provides accuracy, precision, recall, F1-Score, and Kappa with the best value of 99%. In addition, K-Fold Cross Validation is performed on the models that have been evaluated with 89% of accuracy. In the future, the implementation of LCNN in real-world applications is highly expected to be developed. On the other side, there are several aspects in this research regarding the LCNN model on the rice seedlings dataset that can be improved further in future works. First, the training process can be carried out with various epochs for each model so it will have better performance. Secondly, future research can conduct a comparison analysis between the methods used in this research with methods in other studies that have been published. And lastly, parameters such as storage and computational memory in the trained model can be analyzed in future research to improve the lightweight aspect of the architecture model.

Acknowledgement

This work was supported by the Politeknik Negeri Jakarta with grant number 468/PL3.18/PT.00.06/2023.

Reference

[1] S. P. Adhikari, G. Kim, and H. Kim, "Deep neural network-based system for autonomous navigation in paddy field," *IEEE Access*, vol. 8, pp. 71272–71278, 2020, doi: 10.1109/ACCESS.2020.2987642.

[2] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture," *Computer Networks*, vol. 172, May 2020, doi: 10.1016/j.comnet.2020.107148.

[3] L.-H. Li, K.-L. Chung, L.-Q. Jiang, A. Kumar Sharma, and Y.-S. Liu, "The Study of Light-weight YOLOv4 Model for Rice Seedling and Counting."

[4] M. Ashiqul Islam, M. Nymur Rahman Shuvo, M. Shamsojjaman, M. Shahadat Hossain, S. Hasan, and T. Khatun, "An Automated Convolutional Neural Network Based Approach for Paddy Leaf Disease Detection," *Article in International Journal of Advanced Computer Science and Applications*, vol. 12, no. 1, p. 2021, 2021, doi: 10.14569/IJACSA.2021.0120134.

[5] M. Der Yang, H. H. Tseng, Y. C. Hsu, C. Y. Yang, M. H. Lai, and D. H. Wu, "A UAV open dataset of rice paddies for deep learning practice," *Remote Sens (Basel)*, vol. 13, no. 7, Apr. 2021, doi: 10.3390/rs13071358.

[6] X. Ma *et al.*, "Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields," *PLoS One*, vol. 14, no. 4, Apr. 2019, doi: 10.1371/journal.pone.0215676.

[7] H. H. Tseng, M. der Yang, R. Saminathan, Y. C. Hsu, C. Y. Yang, and D. H. Wu, "Rice Seedling Detection in UAV Images Using Transfer Learning and Machine Learning," *Remote Sens (Basel)*, vol. 14, no. 12, Jun. 2022, doi: 10.3390/rs14122837.

[8] M. M. Anuar, A. A. Halin, T. Perumal, and B. Kalantar, "Aerial Imagery Paddy Seedlings Inspection Using Deep Learning," *Remote Sens (Basel)*, vol. 14, no. 2, Jan. 2022, doi: 10.3390/rs14020274.

[9] Y. Zhou, S. Chen, Y. Wang, and W. Huan, "Review of research on lightweight convolutional neural networks," 2020.

[10] Q. Kang, H. Zhao, D. Yang, H. S. Ahmed, and J. Ma, "Lightweight convolutional neural network for vehicle recognition in thermal infrared images," *Infrared Phys Technol*, vol. 104, Jan. 2020, doi: 10.1016/j.infrared.2019.103120.

[11] M. Polsinelli, L. Cinque, and G. Placidi, "A light CNN for detecting COVID-19 from CT scans of the chest," *Pattern Recognit Lett*, vol. 140, pp. 95–100, Dec. 2020, doi: 10.1016/j.patrec.2020.10.001.

[12] L. Yang *et al.*, "An embedded implementation of CNN-based hand detection and orientation estimation algorithm," *Mach Vis Appl*, vol. 30, no. 6, pp. 1071–1082, Sep. 2019, doi: 10.1007/s00138-019-01038-4.

[13] "aipal-nchu/RiceSeedlingDataset: A repository to open rice seedling dataset." https://github.com/aipal-nchu/RiceSeedlingDataset (accessed Oct. 21, 2022).

[14] A. J. Khalil and S. S. Abu-Naser, "Diagnosis of Blood Cells Using Deep Learning," *International Journal of Academic Engineering Research (IJAER)*, vol. 6, pp. 69–84, 2022, Accessed: Oct. 17, 2022. [Online]. Available: www.ijeais.org/ijaer

[15] A. S. M. M. Hasan, F. Sohel, D. Diepeveen, H.

Laga, and M. G. K. Jones, "A survey of deep learning techniques for weed detection from images," *Comput Electron Agric*, vol. 184, p. 106067, May 2021, doi: 10.1016/J.COMPAG.2021.106067.

[16] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep Learning for Hyperspectral Image Classification: An Overview," Oct. 2019, doi: 10.1109/TGRS.2019.2907932.

[17] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data 2021 8:1*, vol. 8, no. 1, pp. 1–74, Mar. 2021, doi: 10.1186/S40537-021-00444-8.

[18] "Tampilan Sistem Pengenalan Benih Padi menggunakan Metode Light Convolutional Neural Network pada Raspberry PI 4 B." https://journal.nurulfikri.ac.id/index.php/jtt/article/view/443/238 (accessed Nov. 09, 2022).

[19] F. Rustam *et al.*, "Wireless Capsule Endoscopy Bleeding Images Classification Using CNN Based Model," *IEEE Access*, vol. 9, pp. 33675–33688, 2021, doi: 10.1109/ACCESS.2021.3061592.

[20] T. S. Gunawan *et al.*, "Development of video-based emotion recognition using deep learning with Google Colab," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2463–2471, Oct. 2020, doi: 10.12928/TELKOMNIKA.V18I5.16717.

[21] A. A. I. A. Maharani, S. S. Prasetiyowati, and Y. Sibaroni, "Classification of Public Sentiment on Fuel Price Increases Using CNN," *Sinkron*, vol. 8, no. 3, Jul. 2023, doi: 10.33395/sinkron.v8i3.12609.

[22] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," *Journal of Applied Informatics and Computing*, vol. 4, no. 1, pp. 45–51, May 2020, doi: 10.30871/JAIC.V4I1.2017.

[23] Z. Wang, M. Hu, and G. Zhai, "Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data," *Sensors (Switzerland)*, vol. 18, no. 4, Apr. 2018, doi: 10.3390/s18041126.