Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Disease Classification on Rice Leaves using DenseNet121, DenseNet169, DenseNet201

Adi Dwifana Saputra[1)*], Djarot Hindarto[2)], Handri Santoso[3)]
[1,2,3)]Universitas Pradita, Serpong, Indonesia
[1)] adi.dwifana@student.pradita.ac.id, [2)]sosbuk@gmail.com, [3)]handri.santoso@pradita.ac.id

Submitted : Nov 7, 2022 | **Accepted** : Nov 7, 2022 | **Published** : Jan 1, 2023

Abstract: Rice is a plant that can grow in the tropics. This plant can produce food that can meet the needs of the people of a country. This plant can grow well if it is cared for properly. If the planting has used good care, such as providing adequate water, adding good fertilizer, it can be ascertained that it will produce a lot of rice fruit after harvesting. This often causes concern if rice growers have given good care but often produce less rice fruit because rice plants are attacked by various diseases. This is what makes the problem, that rice plants are attacked by diseases. Before spraying diseases or pests, farmers should have an understanding of diseases in rice. This makes farmers not wrong in choosing drugs for farmers' rice. It is very vulnerable if farmers do not know about the rice disease. Therefore, it is necessary to observe what types of rice diseases attack rice plants. Observations are not enough just to take pictures with a camera. But it is necessary to carry out further analysis of rice diseases. The presence of information technology is now able to recognize any type. One of the machine learning technologies is able to detect rice diseases. One of these branches of machine learning is deep learning. By using a dataset that focuses on rice disease, the model generated from deep learning training is able to detect rice disease. The purpose of this research is to predict disease in rice leaves using deep learning, namely DenseNet. Training using DenseNet, namely DenseNet121, DenseNet169 and DenseNet201. Accuracy using DenseNet121 reached 91.67%, DenseNet169 reached 90%, and DenseNet201 reached 88.33%. The model training time takes 24 seconds.

Keywords: Rice Leaf Disease Detection; DenseNet121; DenseNet169; DenseNet201; Machine Learning; Deep Learning Training;

INTRODUCTION

Deep learning (Goodfellow et al., 2014) is part of machine learning and has long been presented by several researchers. There is quite a long history in this research journey. First introduced in 1943, where the research "threshold logic to copy human thought processes" discusses the model of deep learning (Muhammad Hammad Saleem, Johan Potgieter, 2016). In 1958 the Foundation of Deep Neural Network (DNN) was also introduced. DNN is a deep learning that modifies an Artificial Neural Network that uses multiple input and output layers. From 1960 to 1985, research continued, in these years it is still developing in backpropagation research. The year 2006 until now began with the discovery of the Convolutional Neural Network (CNN), as in fig 1 deep learning history.

The development of the Convolutional Neural Network (CNN) (Sze et al., 2022) algorithm is very rapid. And several developments have been carried out by many researchers in this field. Research that uses CNN in research, especially computer vision, has produced many algorithms for classifying images in various classes. One of the Keras.io applications is DenseNet. Currently there are DenseNet there are several architectures such as DenseNet121, DenseNet169 and DenseNet201, where each architecture has its own characteristics. DenseNet121 has size 33 MB, accuracy 75% - 93%, parameter 8.1M, deep 242. DenseNet169 has size 57 MB, accuracy 76.2% - 93.2%, parameter 14.3 M, deep 338. DenseNet201 has size 80 MB, accuracy 77.3% - 93.6%, parameter 20.2 M, deep 402.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

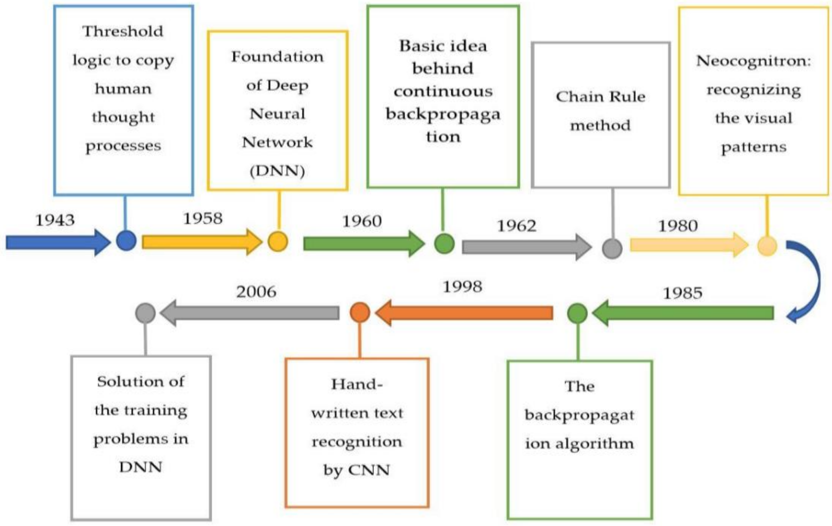


Fig. 1 Deep learning history
Source: (Muhammad Hammad Saleem, Johan Potgieter, 2016)

Rice plants are very important for the needs of the community, especially this plant can thrive in tropical areas such as Indonesia. This plant is a product of basic necessities. If production is disrupted it will cause disaster for the community. Maintenance of rice plants is not enough just to be given water, given fertilizer but preventing or treating disease is part of rice maintenance. Enough water, enough fertilizer, but not caring about disease or pests, causing crop failure. Therefore, it is necessary to take preventive measures against the disease. If the rice plant is already affected by the disease then what is needed is medicine or pesticides. Before using pesticides, the first step is to know or detect diseases in rice plants. This research only discusses the diseased rice leaves. As in fig 2, the healthy rice and rice leaf disease figures are explained. In any field, maintenance and business sustainability make the business grow better. Including in the company, requires business continuity, if the business is in any industry, it is necessary to think about business sustainability for various companies or businesses. Agriculture is an industry in the food sector and must be managed and planned properly. If necessary, state companies take care of rice cultivation. For this reason, rice management companies use IT planning or Enterprise Architecture based on business sustainability (Hindarto et al., 2021) in the rice plant industry.

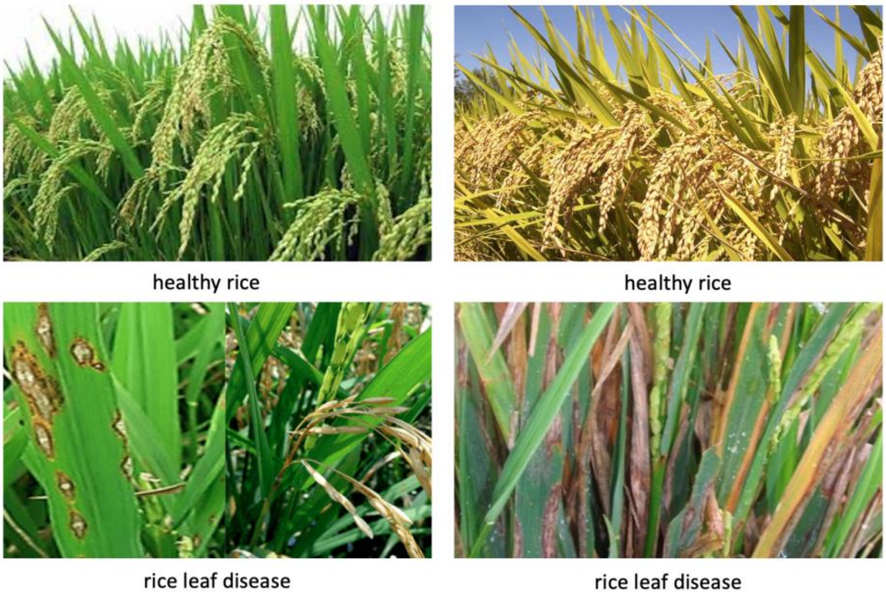


Fig. 2 Rice plants
Source: Google Image

For companies in the agricultural industry, they should also have implemented good information technology, by adding application systems, infrastructure and cyber security. However, the company has yet to strengthen the

*name of corresponding author

SinkrOn JURNAL & PENELITIAN TEKNIK INFORMATIKA
Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

security system, especially with malware security. Because malware can damage information systems and leak company data. The presence of a malware detection and protection system (Hindarto & Santoso, 2022), (Hindarto & Handri Santoso, 2021), (Hindarto, 2022) is very necessary for companies.

The purpose of this research is to detect the disease in rice leaves using DenseNet architecture. Training of this model is done by Pre-Trained. It is hoped that this method can produce a model that has high accuracy and is fast in conducting the training. The dataset needed to conduct training must be pre-processed first, considering that the dataset used must be a ready and clean dataset. Because dirty data will affect the performance of the training. This research raises several questions related to the detection of diseases in rice leaves. Research Question (1), How to get the dataset from this research? Research Question (2), If the dataset has been collected how to produce a training model using deep learning architecture? Research Question (3) What about the performance and speed in conducting model training?

LITERATURE REVIEW

Many studies have discussed deep learning as training in producing deep learning models. These studies are the basis of this research, because there are many research problems that have not been well resolved. This research is not only looking for weaknesses but this research is to complement the researches that will be discussed below.

Discussion about deep learning using DenseNet to detect diseases in rice leaves. The title of his research "Depp Pre-Trained Model Using DenseNet Architecture for Identification of Rice Leaf Diseases" (Faizin et al., 2022). The results of the training model using DenseNet reached 93%. The time needed to do the training model is 31 seconds. Weaknesses do not compare with other DenseNet architectures such as DenseNet121, DenseNet169, and DenseNet201.

Research Classification of Covid-19 patients using efficient fine-tuned deep learning DenseNet model (Bohmrah & Kaur, 2021). This research was conducted with three Dense Architectures such as DenseNet121, DenseNet169 and DenseNet201. The proposal in this research uses Root Mean Square Propagation (RMSprop), so that the accuracy results reach 95.2%. Weaknesses in this research do not explain the training time to become a model using DenseNet.

Research Single-modality and joint fusion deep learning for diabetic retinopathy diagnosis (El-Ateif & Idri, 2022), conducting research using multiple deep learning. The research also uses seven convolutional neural network models (VGG19, ResNet50V2, DenseNet121, InceptionV3, In-ceptionResNetV2, Xception, and MobileNetV2). The results of the model that was trained using DenseNet121 produced a good performance of around 90%. The drawback of this research is that it does not explain the time required for model training.

Research Application of deep learning to identify COVID-19 infection in posteroanterior chest X-rays (Maharjan et al., 2021). Conducting research to identify COVID-19, by conducting training datasets using the Convolutional Neural Network with 6 architectures such as VGG16, DenseNet121, DenseNet201, MobileNet, NasNetMobile and InceptionV3. Dataset from clinical Picture Archiving and Communication System (PACS) database at the National Institutes of Health Clinical Center. It contains 14 disease image labels including atelectasis, cardiomegaly, effusion, infiltration, mass, nodule, pneumonia, pneumothorax, consolidation, edema, emphysema, fibrosis, pleural thickening and hernia. DenseNet121 achieved an AUROC performance of 0.87. Weaknesses do not explain the training time of the dataset into a model.

Research Deep learning of rock microscopic images for intelligent lithology identification: Neural network comparison and selection (Xu et al., 2022). This research uses the ception algorithm, MobileNet_v2, Inception_ResNet_v2, Inception_v3, Densenet121, ResNet101_v2, and ResNet-101. DenseNet121 has an accuracy of about 96.88%. The weakness of this research is not showing the time for training the dataset into a model.

Research Deep transfer learning based classification model for covid-19 using chest CT-scans (LAHSAINI et al., 2021). This research uses algorithms such as DenseNet121, DenseNet201, VGG16, VGG19, Inception Resnet-V2, and Xception. This research proposes a modification of DenseNet201 to conduct research on the Covid-19 dataset. The results of the training achieved an accuracy of 98.18%, Area Under Curve of 98.88%. Weaknesses in this research do not show the training time.

Previous studies using the Convolutional Neural Network algorithm with DenseNet architecture have been discussed and achieved an accuracy of 93% to 98.88%. As for the speed in doing training, there are those that require speeds of up to 31 seconds. In this research, it does not only calculate accuracy but also calculates the speed in detecting or the speed has reached 24 seconds. Where in the previous research above did not calculate speed. Previous research that used speed was a research entitled deep learning using DenseNet to detect diseases in rice leaves and the training time and detection time took 31 seconds. The state of the art in this research performs and calculates the time required for training and detection to reach 24 seconds.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

METHOD

This research uses a deep learning method as a training process to get a model that will detect diseases in rice leaves. Precisely using the Convolutional Neural Network algorithm with DenseNet architecture as the basis for the research. This DenseNet method is considered a simpler method than the ResNet architecture. DenseNet architecture is made to produce accuracy from training (model) because the gradient process can eliminate processes in the neural network network. This process is caused by the distance that is too far in the input layer and the output layer. This process eliminates information before reaching the destination in the Convolution process.

Pre-Trained, banyak riset yang menggunakan model trainingnya dengan pre-trained. Karena pre-trained tidak melakukan training dari awal, sehingga dalam melakukan training pada dataset baru hanya menambahkan model yang sdah dilakukan training sebelumnya. Training any classifier, the first time process is that the initial layer always detects any slashes classified. Therefore, train a new dataset every time you create a neural network. Only the final layer of the network is used and the learning layer is used to detect new layers, only the final layer is used. The pre-training model is trained with a large number of datasets and uses resources that not everyone has the dataset. For example ImageNet for example contains more than 14 million images and 1.2 million are defined as 1000 categories. So the use of the pre-trained model is very useful, because the model used already has a very large training dataset.

DenseNet, is an algorithm of the Convolutional Neural Network which has similarities to the ResNet architecture. The additive method is an additional process used by Resnet to retrieve the previous output in the input process at the layer after the ResNet process. Whereas DenseNet for the input is from all the previous outputs in the process layer after it.

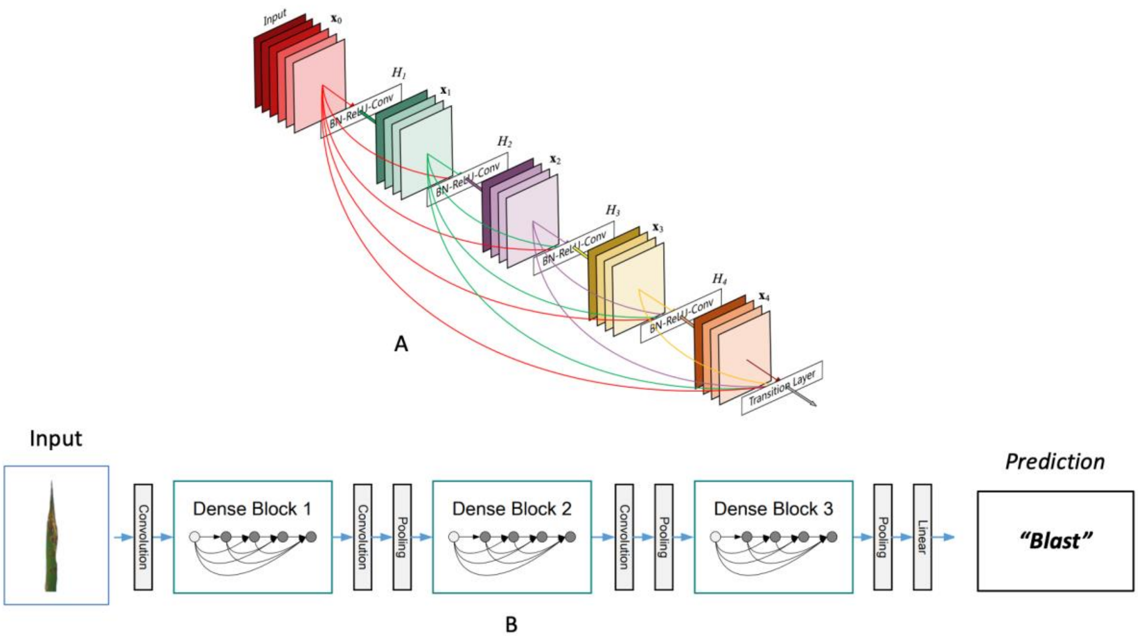


Fig. 3 DenseNet Structure Densely

Sources: DenseNet Structure Densely Connected Convolutional Networks (Huang et al., 2017)

DenseNet Structure dapat dirumuskan sebagai berikut:

$$a^{[1]} = g([a^{[0]}, a^{[1]}, a^{[2]}, \ldots\ldots \ldots\ldots, a^{[l-1]}) \quad (1)$$

Fig 3.A. is a process from DenseNet. Five layers use x as much as 4 as the growth rate and all layer inputs on the map feature are taken. DenseNet requires fewer parameters than traditional CNNs, so the DensNet process does not need to know much about feature maps. The number of parameters in the ResNets architecture is at a very important level in the learning process. DenseNets layers, on the other hand, are very narrow (e.g. 12 filters) because they require an additional set of small size feature maps. The deep learning process in training has the problem of information flow and gradients. The DenseNet architecture can solve the problem by directly accessing the input, the layer of gradients of the original image loss function. F3.B. is the Convolution process and forms Block 1, followed by Convolution then Pooling. The process is continued with the Dense Block 3 process then Covlution and Pooling. The process is continued by Dense Block 3 followed by Pooling and Linear

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

so that it becomes a prediction for rice leaf disease. The DenseNet process is carried out more simply than ResNet with a large number of parameters and is complex in conducting the training process. The input is processed through DenseLayer and combines the input. Then do bn_function in the feature map to produce an output bottleneck. This process can perform computations efficiently. In the final stage, the convolution process is carried out and produces a new feature of size K as the growth rate.

Here DenseNet121 = 5 + (6 + 12 + 24 + 16) * 2 = 121, with the explanation that
5 – Convolution and Pooling Layer.
3 – Trasition Layers (6,12,24).
1 – Classification Layer (16).
2 – DenseBlock (1x1 and 3x3 conv).

Optimizer='adam', Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to iteratively update the weights based on the training data (Kingma & Ba, 2015). Adam can be said to be a combination of RMSprop and Stochastic Gradient Descent with momentum. Adam is an adaptive learning rate method, where Adam calculates individual learning rates for different parameters. The name “Adam” comes from “adaptive moment estimation” because Adam uses the first and second moment gradient estimates to adapt the learning rate for each neural network weight.
The advantages of this Adam optimization are:

- Easy to apply.
- Computationally efficient.
- Small memory requirements.
- Not much different from gradient with diagonal scale.
- Appropriate in terms of data and/or parameters that have major problems.
- Suitable for gradients with high noise.

How Does Adam Work? Adam differs from classical stochastic gradient descent. Stochastic gradient descent (Fjellström & Nyström, 2022), (Li et al., 2022) maintains a single learning rate (alpha) for all weight updates and the learning rate does not change during training. The learning rate is maintained for each network weight (parameter) and is adapted separately as learning progresses. The method calculates the individual adaptive learning rates (Pepper et al., 2022), (Kotsyuba et al., 2022) for different parameters of the first and second moment estimates of the gradient.

Adam as a combination of the advantages of two extensions of stochastic gradient descent. In particular:

- Adaptive Gradient Algorithm (AdaGrad) which maintains a per-parameter learning rate which improves performance on problems with diffused gradients.
- Root Mean Square (RMSProp) propagation which also maintains an adapted per-parameter learning rate based on the average of the latest gradient magnitude for the weights (i.e. how fast it changes). That is, the algorithm works well on non-constant problems such as noise.
- In addition to adapting the learning rate parameter based on the average of the first moment (mean) as in RMSProp, Adam also makes use of the second moment average of the gradient (uncentralized variance).

Adam is a popular algorithm in the field of deep learning because Adam can achieve good results quickly. In his original paper, Adam was shown empirically to show that convergence met the expectations of theoretical analysis. Adam was applied to the logistic regression algorithm on the MNIST digit recognition and IMDB sentiment analysis dataset, the Multilayer Perceptron algorithm on the MNIST dataset and Convolutional Neural Networks on the CIFAR-10 image recognition dataset.

Performance, Most of the accuracy to evaluate the accuracy of a model, then used a confusion matrix in evaluating the performance of the classification model. In the confusion matrix the predictions of the model along the x-axis and labels along the y-axis. True Positive is the number of samples in rice leaf disease as true rice leaf disease. True Negative is the number of negative samples in rice leaf disease. False Positive is the number of negative samples that are incorrectly identified as positive and False Negative (FN) is the number of positive samples that are incorrectly identified as negative. Acc is formulated as follows.

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

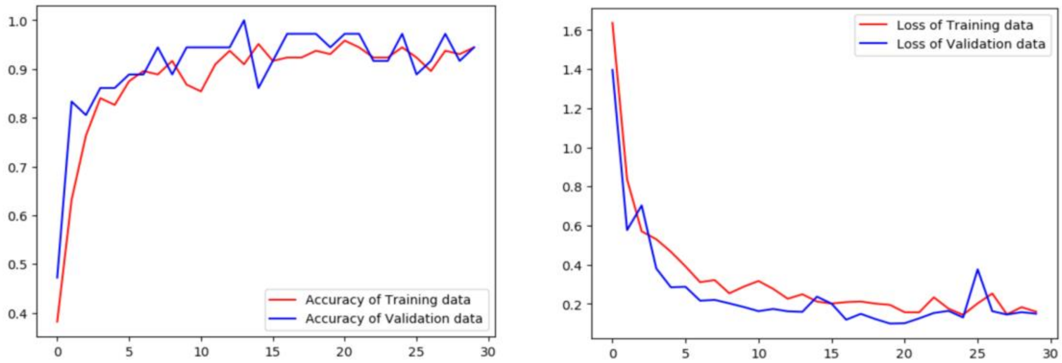
Detection accuracy is not the only parameter that is very important. Speed in computing the training model is a performance metric to evaluate the model. In engineering, rapid identification increases the efficiency of the applied model. FPS needs to be compared under the following conditions. Hardware is in the same condition. Faster to identify when the FPS value is higher.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

RESULT

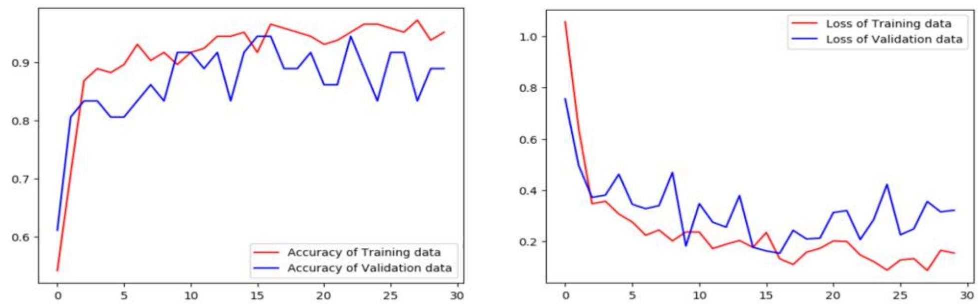
The results of research using the Convolutional Neural Network algorithm with DenseNet121, DenseNet169 and DenseNet201 architectures, show good accuracy results. Where the average results reach 90% and requires a fast training time of about 24 seconds to detect diseases in rice leaves. The reason for using DenseNet is because this architecture is able to improve the performance of the Convolutional Neural Network, in addition to the accuracy obtained, it also measures the time required to recognize objects.



	precision	recall	f1-score	support
0	0.88	1.00	0.94	15
1	1.00	1.00	1.00	13
2	1.00	0.75	0.86	8
accuracy			0.94	36
macro avg	0.96	0.92	0.93	36
weighted avg	0.95	0.94	0.94	36

Fig. 4. Accuracy and Loss DenseNet121
Source: Researcher Propeerty

Fig.4 is the result of training using the Convolutional Neural Network (CNN) Algortima with DenseNet121 architecture. The results of training using Pre-Trained produce an average accuracy of 94%. According to this research is a high yield for DenseNet121 architecture.



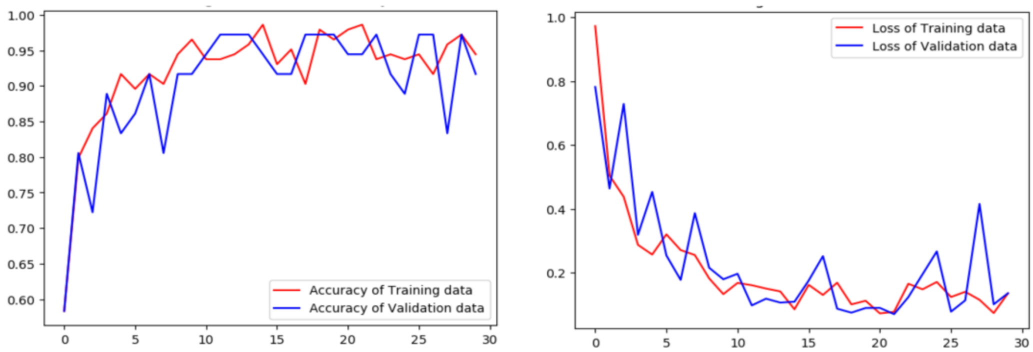
	precision	recall	f1-score	support
0	0.93	0.93	0.93	15
1	1.00	0.85	0.92	13
2	0.70	0.88	0.78	8
accuracy			0.89	36
macro avg	0.88	0.88	0.88	36
weighted avg	0.91	0.89	0.89	36

Fig. 5. Accuracy and Loss DenseNet169
Source: Researcher Propeerty

Fig.5 is the result of training using the Convolutional Neural Network (CNN) Algortima with DenseNet121 architecture. The results of training using Pre-Trained produce an average accuracy of 89%. According to this research, this is not a very high result for the DenseNet169 architecture.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X




```
              precision    recall  f1-score   support

           0       0.88      1.00      0.94        15
           1       1.00      0.92      0.96        13
           2       0.86      0.75      0.80         8

    accuracy                           0.92        36
   macro avg       0.91      0.89      0.90        36
weighted avg       0.92      0.92      0.92        36
```

Fig. 6. Accuracy and Loss DenseNet201
Source: Researcher Propeerty

Fig.6 is the result of training using the Convolutional Neural Network (CNN) Algortima with DenseNet201 architecture. The results of training using Pre-Trained produce an average accuracy of 92%. According to this research is a high yield for DenseNet201 architecture.

Table 1. Summary Accuracy, Precission, Recall, and F1-Score

No	Performance	DenseNet121	DenseNet169	DenseNet201
1	Accuracy	94%	89%	92%
2	Precission	96%	87,67%	91,33%
3	Recall	91,66%	88,67%	89%
4	F1-Score	93,33%	87,67%	90%
5	Support	12	12	12

Table 1 is a summary of the results of training using Pre-Trained from the Convolutional Neural Network (CNN) algorithm with DenseNet121, DenseNet169, and DenseNet201 architectures..

DISCUSSIONS

How to get the dataset from this research? (RQ 1). The dataset in this research uses a dataset from the public dataset, namely Kaggle and a dataset from the Google image dataset. The addition of the dataset will make the training model more accurate when compared to the public dataset without the addition. This is where the importance of training a model. Research Question (RQ 2), If the dataset is already collected how to generate a training model using deep learning architecture? Convolutional Neural Networks with DenseNet architecture are used as the training model. This research does not use training from the beginning, but uses Pre-Trained, so it does not train the dataset from the beginning. Research Question (RQ 3). What about the performance and speed in conducting model training?

CONCLUSION

The conclusion of this research, how to detect disease in rice leaves has been carried out using the Convolutional Neural Network algorithm with DenseNet architecture. The ones used in this research are DenseNet121, DenseNet169 and DenseNet201. Accuracy using DenseNet121 reached 91.67%, DenseNet169 reached 90%, and DenseNet201 reached 88.33%. The model training time takes 24 seconds. Accuracy in this research is not very important, but accuracy without being supported by speed in recognizing objects or diseases in rice leaves cannot be said to be perfect. Research using classifications, besides accuracy, should also consider the speed factor. Much of the research does not focus on training time. So in this study, training time is the focus, but not the main focus in this research. Training time is used as the focus of this research because training time becomes an obstacle if the required training time is too long.

REFERENCES

Bohmrah, M. K., & Kaur, H. (2021). Classification of Covid-19 patients using efficient Fine-tuned Deep

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 7, Issue 1, January 2023
DOI : https://doi.org/10.33395/sinkron.v8i1.11906
e-ISSN : 2541-2019
p-ISSN : 2541-044X

learning DenseNet Model. *Global Transitions Proceedings*, 0–14. https://doi.org/10.1016/j.gltp.2021.08.003

El-Ateif, S., & Idri, A. (2022). Single-modality and joint fusion deep learning for diabetic retinopathy diagnosis. *Scientific African*, *17*, e01280. https://doi.org/10.1016/j.sciaf.2022.e01280

Faizin, A., Arsanto, A. T., Musa, A. R., Informatika, P. T., Pasuruan, U. Y., & Blight, B. (2022). *DEEP PRE-TRAINED MODEL MENGGUNAKAN ARSITEKTUR DENSENET*. *6*(2), 615–621.

Fjellström, C., & Nyström, K. (2022). Deep learning, stochastic gradient descent and diffusion maps. *Journal of Computational Mathematics and Data Science*, *4*(June), 100054. https://doi.org/10.1016/j.jcmds.2022.100054

Goodfellow, I., Mehdi, M., Bing, X., & David, W. (2014). *Generative Adversarial Nets*.

Hindarto, D. (2022). Perbandingan Kinerja Akurasi Klasifikasi K-NN, NB dan DT pada APK Android. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, *9*(1), 486–503. https://doi.org/10.35957/jatisi.v9i1.1542

Hindarto, D., & Handri Santoso. (2021). Android APK Identification using Non Neural Network and Neural Network Classifier. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, *5*(2), 149–157. https://doi.org/10.29303/jcosine.v5i2.420

Hindarto, D., Indrajit, R. E., & Dazki, E. (2021). Sustainability of Implementing Enterprise Architecture in the Solar Power Generation Manufacturing Industry. *Sinkron*, *6*(1), 13–24. https://jurnal.polgan.ac.id/index.php/sinkron/article/view/11115

Hindarto, D., & Santoso, H. (2022). PERFORMANCE COMPARISON OF SUPERVISED LEARNING USING NON-NEURAL NETWORK AND NEURAL NETWORK. *Janapati*, *11*, 49–62.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-Janua*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Kotsyuba, I., Shikov, A., Romanov, K., Galperin, M., Kudriashov, A., & Zhukova, T. (2022). The development of the recommendatory resource for the adaptive learning of transport philosophy. *Transportation Research Procedia*, *63*, 600–606. https://doi.org/10.1016/j.trpro.2022.06.053

LAHSAINI, I., EL HABIB DAHO, M., & CHIKH, M. A. (2021). Deep transfer learning based classification model for covid-19 using chest CT-scans. *Pattern Recognition Letters*, *152*, 122–128. https://doi.org/10.1016/j.patrec.2021.08.035

Li, Q., Xiong, D., & Shang, M. (2022). Adjusted stochastic gradient descent for latent factor analysis. *Information Sciences*, *588*, 196–213. https://doi.org/10.1016/j.ins.2021.12.065

Maharjan, J., Calvert, J., Pellegrini, E., Green-Saxena, A., Hoffman, J., McCoy, A., Mao, Q., & Das, R. (2021). Application of deep learning to identify COVID-19 infection in posteroanterior chest X-rays. *Clinical Imaging*, *80*, 268–273. https://doi.org/10.1016/j.clinimag.2021.07.004

Muhammad Hammad Saleem, Johan Potgieter, K. M. A. (2016). Plant Disease Detection and Classification by Deep Learning. *Nature*, *29*(7553), 1–73. http://deeplearning.net/

Pepper, N., Crespo, L., & Montomoli, F. (2022). Adaptive learning for reliability analysis using Support Vector Machines. *Reliability Engineering and System Safety*, *226*(June), 108635. https://doi.org/10.1016/j.ress.2022.108635

Sze, E., Santoso, H., & Hindarto, D. (2022). *Review Star Hotels Using Convolutional Neural Network*. *7*(1), 2469–2477.

Xu, Z., Ma, W., Lin, P., & Hua, Y. (2022). Deep learning of rock microscopic images for intelligent lithology identification: Neural network comparison and selection. *Journal of Rock Mechanics and Geotechnical Engineering*, *14*(4), 1140–1152. https://doi.org/10.1016/j.jrmge.2022.05.009

*name of corresponding author