

Detection of Disease and Pest of Kenaf Plant Based on Image Recognition with VGGNet19

Diny Melsye Nurul Fajri ^{a,1*}, Wayan Firdaus Mahmudy ^{a,2}, Titiek Yulianti ^{b,3}

^a Faculty of Computer Science, Brawijaya University
8th Veteran Road Malang 65145, Jawa Timur, Indonesia

^b Indonesian Research Institute for Sweetener and Fiber Crops (IRISFC)
Karangploso km 4, PO.Box 199, Kabupaten Malang, Jawa Timur, Indonesia

¹ dimelnf@gmail.com*; ² wayanfm@ub.ac.id; ³ tyuliant@gmail.com

* corresponding author

ARTICLE INFO

Article history:

Received 14 January 2021

Revised 9 May 2021

Accepted 13 August 2021

Published online 17 August 2021

Keywords:

convolutional neural network

image recognition

kenaf

neural network

VGGNet19

ABSTRACT

One of the advantages of Kenaf fiber as an environmental management product that is currently in the center of attention is the use of Kenaf fiber for luxury car interiors with environmentally friendly plastic materials. The opportunity to export Kenaf fiber raw material will provide significant benefits, especially in the agricultural sector in Indonesia. However, there are problems in several areas of Kenaf's garden, namely plants that are attacked by diseases and pests, which cause reduced yields and even death. This problem is caused by the lack of expertise and working hours of extension workers as well as farmers' knowledge about Kenaf plants which have a terrible effect on Kenaf plants. The development of information technology can be overcome by imparting knowledge into machines known as artificial intelligence. In this study, the Convolutional Neural Network method was applied, which aims to identify symptoms and provide information about disease symptoms in Kenaf plants based on images so that early control of plant diseases can be carried out. Data processing trained directly from kenaf plantations obtained an accuracy of 57.56% for the first two classes of introduction to the VGGNet19 architecture and 25.37% for the four classes of the second introduction to the VGGNet19 architecture. The 5×5 block matrix input feature has been added in training to get maximum results.

This is an open access article under the CC BY-SA license
(<https://creativecommons.org/licenses/by-sa/4.0/>).

I. Introduction

Kenaf plant fiber is one of the basic ingredients for environmental management products that have become the center of global attention. This fiber is intended to replace the consumption of plastic materials used in car interiors at popular automotive companies [1]. This causes the export value of raw materials to increase because they have to meet production needs. A problem was encountered in one of Kenaf's plantations, where some plants died, and the harvest failed. The cause of this crop failure was the kenaf plant which was attacked by diseases and pests. Farmers do not know the plants affected by this disease because of a lack of counseling from experts, which can be fatal. With the development of information technology systems, this can be overcome by detecting symptoms of disease/pests in kenaf plants from an early age to avoid the consequences of crop failure. One of the methods offered is to detect symptoms of disease/pests in kenaf plants through pictures.

A machine carries out image recognition then the machine can make decisions based on the trained image. This machine-driven decision-making system is called machine learning [2]. One application of the machine learning method has been carried out by Saragih in the decision-making system for the classification of jatropha plant disease with an accuracy of 60.61% [3]. This result is categorized as quite good because it can achieve accuracy above 50%. The data used in this study [3] is a scoring of disease types in jatropha, which is given by the expert directly. The more data that is trained, the more complex the machine learning system will be as the data described in one image. One image can include hundreds of thousands or even millions of data that must be processed. Deep complex machine

learning to process a lot of data, such as data in images, is called deep learning. The use of deep learning is often found in everyday cellphone applications such as google photos, QR scanners, even for biometric security systems to face unlock [4]. Several studies using deep learning [5][6][7][8] were able to get 90% -95% accuracy results, which indicates that learning from deep learning is better than machine learning.

Convolutional Neural Network (CNN) is the primary method for classifying images in machine learning. This method consists of some layers, i.e., convolutional layer, subsampling layer, and fully connected layer. The arrangement of these layers is called architecture. CNN has some architecture. One of them is Visual Geometry Group (VGGNet) 19, which focused on the effect of CNN depth on its accuracy [9]. Several studies using the concept of VGG architecture get a reasonably high accuracy value; One of the studies conducted by Khrisne and Suyadna in their research was to identify the types of Indonesian herbs and spices [10]. The objects used have many similarities in models and shapes, so that the introduction made using the VGG architecture reaches 70% in 3527 images which are divided into 27 classes. Other studies also show that the VGGnet architecture is also good enough to recognize types of diseases. The research was conducted using three different types of architecture, namely StridedNet, LeNet, and VGGNet. The results show that the VGGnet architecture has a higher accuracy value than other architectures, reaching 95.40%, while the LeNet model with 93.65% and StrideNet figures reaches 90.10% [11]. Based on field conditions and supported by current technological advances, almost all farmers have used smartphones, which means this is very helpful for farmers to be able to control the conditions of their Kenaf plants so that they can receive early supervision to prevent crop failures caused by pests and diseases through their smartphones. In information technology, this is quite a challenge to help facilitate farmers and save timing.

This research will introduce the types of pests and diseases on kenaf plants using image-based deep learning techniques. This research claims to continue previous research that has been carried out by Fajri [12], who instilled a similar method on the same object. Changes made in this study were to provide training for horizontal 5×5 matrix calculations on the image data in the input model that was used to carry out more effective, efficient, and maximum recognition training. The final result will undoubtedly change if the changes are made in the input model. The image data used in this study are the same as previous studies included in the research boundaries due to the plantation conditions. This research is dedicated to helping plantation farmers, especially Kenaf plants, through a technological approach to an artificial intelligent decision-making system in the field of information technology and also provide results based on research, whether the method used is the right method for detection and how accurate the results of the method that have been designed.

II. Methods

Some of the methods that underlie this research include deep learning, CNN and its architecture, and the proposed method to get good results in making a system to detect symptoms of pests and diseases of kenaf plants.

A. Object Recognition

The object used in this study is the kenaf plant. There are various kinds of kenaf leaf morphology, such as unlobed (figure 1a), partially lobed (figure 1b), and deeply lobed (figure 1c). For deeply lobed, there are 5-7 lobed and have serrations. The cross-section of the kenaf leaf is shown in Figure 1.

On healthy leaves, the leaves are fresh green without any reddish or yellowish to brownish spots. Figure 2 shows healthy kenaf leaf; fresh leaves stretch without any curved parts are categorized under the criteria of "sehat".

Many types of diseases can attack kenaf plants, including Fusarium wilt, root rot, bacterial wilt, leaf blight, and anthracnose [13]. The most frequently encountered is leaf blight and matches on the object used in this study, on leaves. The leaf surface looks brownish, and there are black spots with fungal pycnidia in the middle of the spots [13]. Leaf blight is shown in Figure 3.

The pests that attack also give signs of symptoms on the leaves in leaf edges that start to turn yellow until they appear reddish and curl inward. Symptoms like this are due to the attack of the *Amrasca bigutulla* pest called *sundapteryx* disease on kenaf leaves. The symptoms of *sundapteryx* can be seen in Figure 4.

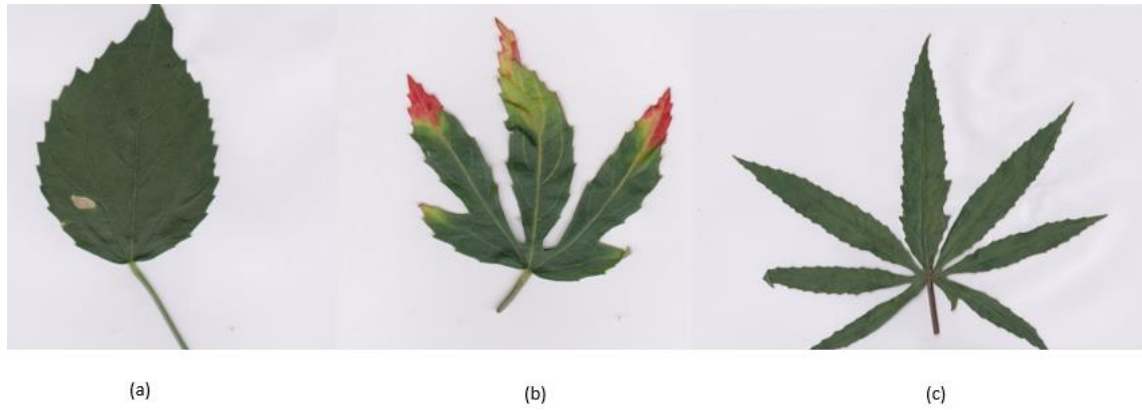


Fig. 1. Kenaf's leaf cross section; (a) unlobed, (b) partially lobed, and (c) deeply lobed



Fig. 2. Kenaf's healthy leaf



Fig. 3. Leaf blight disease on kenaf leaf

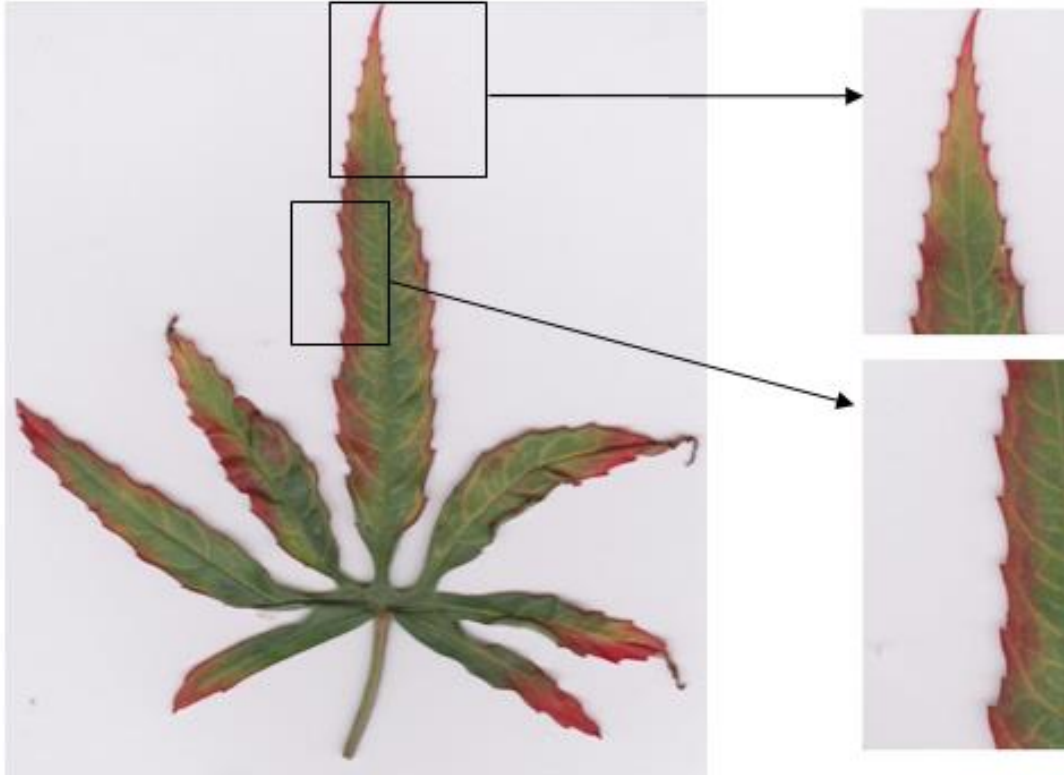


Fig. 4. Sundapteryx Bigutulla on Kenaf Leaf

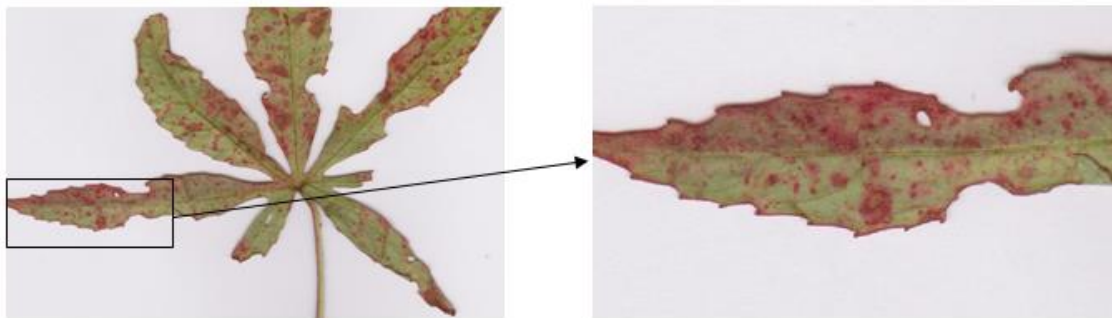


Fig. 5. Tungau affected in Kenaf Leaf

Another symptom that affects kenaf leaves is reddish patches that spread over almost all over the leaf surface and cause the leaves to dry out. The dried leaves will fall off, causing the plant to die. The cause of the appearance of this symptom is mites (*Tungau urticae*). *T. urticae* is usually found in areas with temperatures around 30 °C with low humidity (dry) [14]. Symptoms of mites are shown in Figure 5.

B. Deep Learning

Deep learning is one of the algorithms of neural network algorithms. Input to learning is metadata which is then processed by calling hidden layers to produce the output value. The advantage of this deep learning technique is that it has a unique feature to perform automatic feature extraction. This indicates that deep learning can get a differentiating/unique feature relevant to the problem [15]. The way the engineering works is quite complex because it focuses on network architecture and its optimal procedural features. Deep learning can self-tun and select the optimal training model without requiring excess information [16]. The use of deep learning for learning with image objects can be done with a Convolutional Neural Network.

C. Convolutional Neural Network

Convolutional Neural Network is the development of a neural network designed to process two-dimensional data in images and audio. Yann Le Chun developed CNN in 1990 through handwriting and number recognition [17]; this method won the ImageNet Large Scale Visual Recognition Challenge competition presented by Alex Krizhevsky with 1.3 million high-resolution images LSVRC-2010 ImageNet training set into the 1000 different classes. The basic concept of CNN is that it receives two-dimensional data and then passes it on to the next layer to be processed into an output. For every data that enters the layer, a linear operation will be carried out with a determined weight value then transformed using a non-linear operation called the activation function.

1) Activation function

The activation function acts as a determinant of neurons. Neurons are determined to be active or inactive based on their weight. One type of activation function that is widely used is the Rectified Linear Unit (ReLU). ReLU is applied by making a threshold at zero, or it can be translated as follows, if $x \leq 0$, then $x = 0$, and if $x \geq 0$, then $x = x$ [18]. The equation can be seen in (1) and the curve shown in Figure 6.

$$f(x) = \max(0, x) \quad (1)$$

2) Dropout regularization

A neural network regularization technique means reducing the number of inactive neurons by removing them. The contribution of the wasted neurons will be temporarily suspended. The selection of neurons is made to reduce the risk of overfitting [19]. The dropout technique is applied in Figure 7.

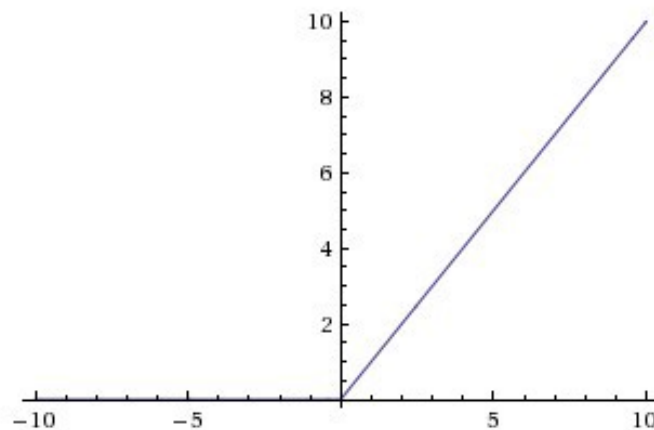


Fig. 6. ReLU activation function

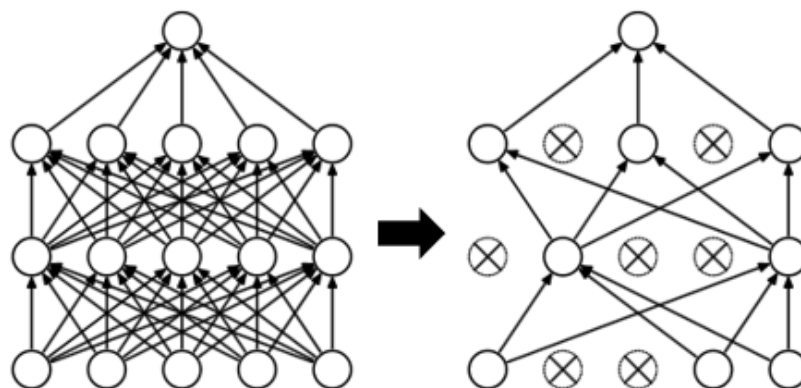


Fig. 7. Dropout technique

3) CNN architecture

Layers and neurons cannot be defined by precise rules and are subject to different treatments on different data types [20]. Several layers can be combined to get more accurate results and better processing performance. The main layers on CNN are as follows:

a) Convolution layer

The convolution layer is the primary layer of CNN architecture, has a convolution process from the previous layer. Convolution implements the function of a kernel at all possible offsets. The kernel moves from the top left corner to the bottom right corner. The purpose of this process is to extract features from the image. The convolution result is a linear transformation of the input according to the spatial information of the data. CNN convolutional operations are shown in Figure 8.

b) Subsampling layer

The subsampling layer is a layer that reduces the size of the image. This layer aims to increase the position invariance of the features. Several techniques in the subsampling layer method include Average Pooling, K Max Pooling, and Max Pooling. Average pooling is to find the average value of each dimension, then for K Max pooling is to find the most considerable K value for each dimension then combined, and the subsampling technique that is often used is max pooling. Max Pooling works by breaking the output from the convolution layer into smaller parts and taking the highest value from each part to compose the reduced image matrix. The concept of the pooling technique can be seen in Figure 9.

c) Fully connected layer

A fully Connected Layer is a layer commonly used in MLP, which aims to transform data dimensions so that data can be classified linearly. This layer is located at the very end of the network. Each neuron that is passed on to the Fully Connected Layer must be one-dimensional since data can lose spatial information and is not reversible.

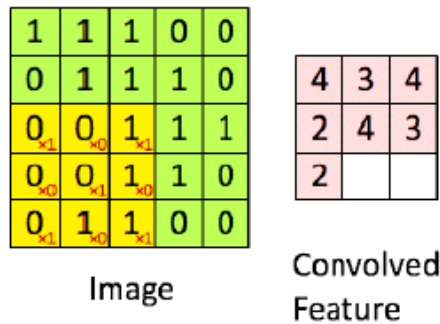


Fig. 8. Convolution process in CNN

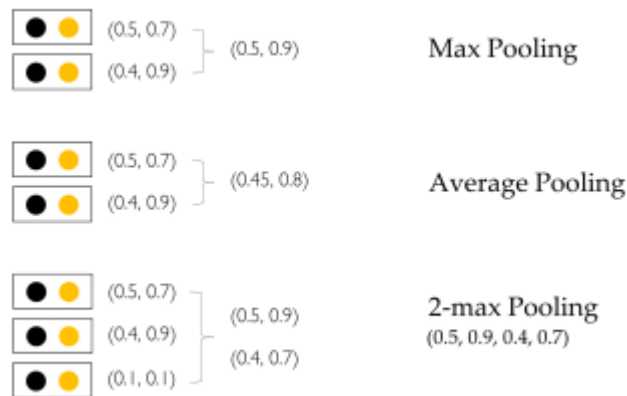


Fig. 9. Pooling concept

D. VGGNet Architecture

An input to perform with CNN uses 224×224 RGB in the pixel range 0 to 255 and reduces the average value of the calculated image in the training sequence. These images then pass through the convolutional layers and the connected layers. The VGGnet architecture has a smaller 3×3 filter. It has the same receptive plane as if it had one layer of 7×7 convolution [9]. An illustration of the architecture can be seen in Figure 10.

E. Proposed Methods

In the type recognition training, the target was four outputs, namely “Hawar Daun”, “Sehat”, “Sundapteryx” and “Tungau”. The basic architectural modeling used in this training uses VGG Net 19. This is another variation of VGGNet. The layers designed in this training consist of input layers \rightarrow 2 blocks for layer 2 convolution and max-pooling \rightarrow 3 blocks for layer 4 convolution and max-pooling \rightarrow flatten layer \rightarrow 2 blocks of dense layer and dropout \rightarrow dense layer. The architectural modeling design of this system can be seen in Figure 11.

An input model involves the matrix calculation from the image data to make a more significant introduction. This matrix calculation is done by creating a 5×5 pixel matrix block from the image data. Then after the block division is carried out, the matrix of each block will be calculated as the average value. The average calculation value in the image matrix used the Numpy module in Python, namely the method `matrix.mean()` and implemented in the source code shown in Figure 12. This calculation is carried out horizontally. This training aims to increase the introduction of knowledge that represents the accuracy of the result.

After the CNN training process has been implanted, the following process is the classification process. This process determines the results of the classification based on previous training. The classification flow begins with entering image data and then processed according to CNN training, then scanning is carried out using a 96×96 sliding window. During the scanning process, the system will detect whether the image is a leaf image or not. If it's not a leaf, the process will be terminated after issuing the "Bukan Daun" result, and the window will give you the option to re-select the image. If the inserted image is a leaf image, the results will be saved, then proceed with the classification process by selecting the range using the non-max suppression algorithm, which gives the final decision of the scanned image then the type of disease results will be displayed based on the percentage.

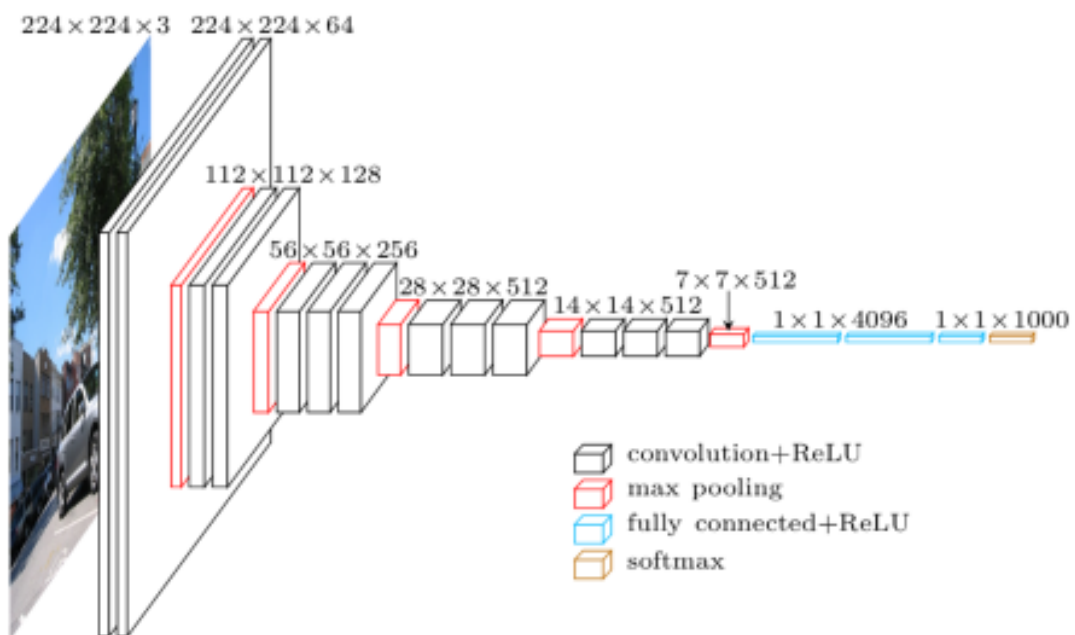


Fig. 10. VGG Architecture

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 96, 96, 3)	0
block1_conv1 (Conv2D)	(None, 96, 96, 64)	1792
block1_conv2 (Conv2D)	(None, 96, 96, 64)	36928
block1_pool (MaxPooling2D)	(None, 48, 48, 64)	0
block2_conv1 (Conv2D)	(None, 48, 48, 128)	73856
block2_conv2 (Conv2D)	(None, 48, 48, 128)	147584
block2_pool (MaxPooling2D)	(None, 24, 24, 128)	0
block3_conv1 (Conv2D)	(None, 24, 24, 256)	295168
block3_conv2 (Conv2D)	(None, 24, 24, 256)	590080
block3_conv3 (Conv2D)	(None, 24, 24, 256)	590080
block3_conv4 (Conv2D)	(None, 24, 24, 256)	590080
block3_pool (MaxPooling2D)	(None, 12, 12, 256)	0
block4_conv1 (Conv2D)	(None, 12, 12, 512)	1180160
block4_conv2 (Conv2D)	(None, 12, 12, 512)	2359808
block4_conv3 (Conv2D)	(None, 12, 12, 512)	2359808
block4_conv4 (Conv2D)	(None, 12, 12, 512)	2359808
block4_pool (MaxPooling2D)	(None, 6, 6, 512)	0
block5_conv1 (Conv2D)	(None, 6, 6, 512)	2359808
block5_conv2 (Conv2D)	(None, 6, 6, 512)	2359808
block5_conv3 (Conv2D)	(None, 6, 6, 512)	2359808
block5_conv4 (Conv2D)	(None, 6, 6, 512)	2359808
block5_pool (MaxPooling2D)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 1024)	4719616
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 256)	262400
dropout_2 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 4)	1028
=====		
Total params: 25,007,428		
Trainable params: 19,141,892		
Non-trainable params: 5,865,536		
=====		
Found 227 images belonging to 4 classes.		
Found 227 images belonging to 4 classes.		
{'HAWAR_DAUN': 0, 'SEHAT': 1, 'SUNDAPTERYX': 2, 'TUNGAU': 3}		

Fig. 11. VGGNet 19 architecture

```
matrix5x5 = test_image[r:r+windowsize_r,c:c+windowsize_c]
meanmatrix5x5 = matrix5x5.mean()
listnewimageX[len(listnewimageX)-1].append(meanmatrix5x5)
```

Fig. 12. Matrix mean source code

III. Results and Discussions

A. Leaf Recognition

Testing accuracy in architectural modeling in leaf recognition training, namely leaf shape recognition, gets an accuracy value of 57.56%. The average value of a maximum of two classes: *precision*, *recall*, and *f₁ score* amount 51%, and the average weight value is 72%. After the model is formed, the results of the value plot graph can be seen in [Figure 13](#). Leaf recognition training has an epoch value of 100. The accuracy value obtained from an epoch of 100 can be seen in [Table 1](#).

B. Disease Classification

Based on the introductory training that has been designed with such modeling, testing the accuracy value for the classification results is carried out by testing similar to the previous training. Testing accuracy in disease recognition training, namely introducing kenaf plant disease classification, gets an accuracy value of 25.37%. The maximum average value between the four classes, namely: recall is 30%, precision and f1 score is 29%, and the average weight value is 32%. The graph has increased in accuracy and decreased data loss, so this is good enough to continue at a later stage. This can be seen in [Figure 14](#). Disease recognition training has an epoch value of 100. The accuracy value obtained from an epoch of 100 can be seen in [Table 2](#).

C. Feature Input Matrix Block

Based on the input feature introduction training that has been designed with such modeling, testing the accuracy value for the classification results is carried out by testing similar to the previous training. For accuracy testing, horizontal input training gets an accuracy value of 24%. The maximum average value between the four classes, namely: recall, precision, and f1 score of 28%, can be seen in [Figure 15](#). The results of testing for disease recognition can be seen in [Table 3](#).

From the total sample data, 200 data are divided into 4, namely 50 data on leaf blight, 50 Sundapteryx, 50 mites, and 50 Healthy. The machine can correctly recognize 106 types of leaf disease data from 200 tested data. The machine also experienced an error in introducing 2 leaf blight data, 17

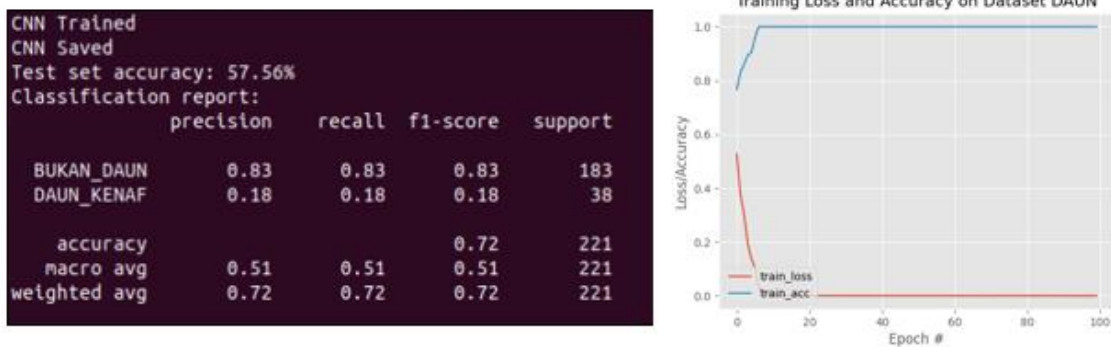


Fig. 13. Accuracy and plotting architectural model



Fig. 14. Accuracy and classification charts

Table 1. Leaf recognition training accuracy with the epoch approach is up to 100

Epoch	Train Accuracy	Train Loss	Time
1	0.7672	0.525	52s
2	0.8307	0.3751	57s
3	0.8594	0.3038	59s
4	0.8942	0.1964	59s
5	0.9032	0.1389	59s
...			
96	1.0000	0.00008467	64s
97	1.0000	0.000091677	63s
98	1.0000	0.000078224	63s
99	1.0000	0.000098397	64s
100	1.0000	0.000060537	63s

Table 2. Disease recognition training accuracy with the epoch approach is up to 100

Epoch	Train Accuracy	Train Loss	Time
1	0.3231	1.6566	53s
2	0.3393	1.3856	65s
3	0.4036	1.3153	50s
4	0.4564	1.2401	60s
5	0.4911	1.1098	70s
...			
96	1.0000	0.0179	67s
97	0.9949	0.0222	67s
98	0.9949	0.0157	67s
99	0.9955	0.0135	77s
100	1.0000	0.0144	67s

Table 3. Correction of disease recognition system

No	Class	Actual Data	System Prediction
1	Leaf Blight (Hawar Daun)	50	48
2	Tungau	50	0
3	<i>Sundapteryx</i> Bigutulla	50	33
4	Healthy (Sehat)	50	25
Total		200	106

Sundapteryx data, 50 mite data, and 25 healthy data. Based on calculations for disease classification, the accuracy value is 53%. The object with the correct prediction is shown in Figure 16.

In Figure 16, the actual data show that these leaves are affected by symptoms that refer to the category of leaf blight disease. This system shows the percentage of classification in the category of leaf blight by 100%.

In Figure 17, the system shows the kenaf plant species in the healthy category by 75%. The results show that 75% is the highest number from other categories, so it can be concluded that the figure falls into the category: Healthy. The actual data also shows that the image is in the healthy category. For the detection of a wrong disease recognition object, it can be seen in Figure 18.

In Figure 18, the actual data shows that the image is in the *Sundapteryx bigutulla* category, while the prediction of the system states that the image is a Mite category with a percentage of 42.1%. Prediction error in this image is due to the training data image factor, which has high similarity between categories. The similarities can be seen in the histogram image shown in Figure 19. The histogram shows the spread of the pixel intensity values of the image. Figure 19 is a histogram image in the *Sundapteryx* category and mites that are similar, making it one of the causes of prediction errors made by the system.

Test set accuracy: 24.44%

Classification report:

	precision	recall	f1-score	support
HAWAR_DAUN	0.33	0.33	0.33	201
SEHAT	0.17	0.17	0.17	69
SUNDAPTERYX	0.37	0.37	0.37	236
TUNGAU	0.22	0.22	0.22	142
accuracy			0.31	648
macro avg	0.27	0.27	0.27	648
weighted avg	0.31	0.31	0.31	648

Fig. 15. Horizontal input feature training accuracy

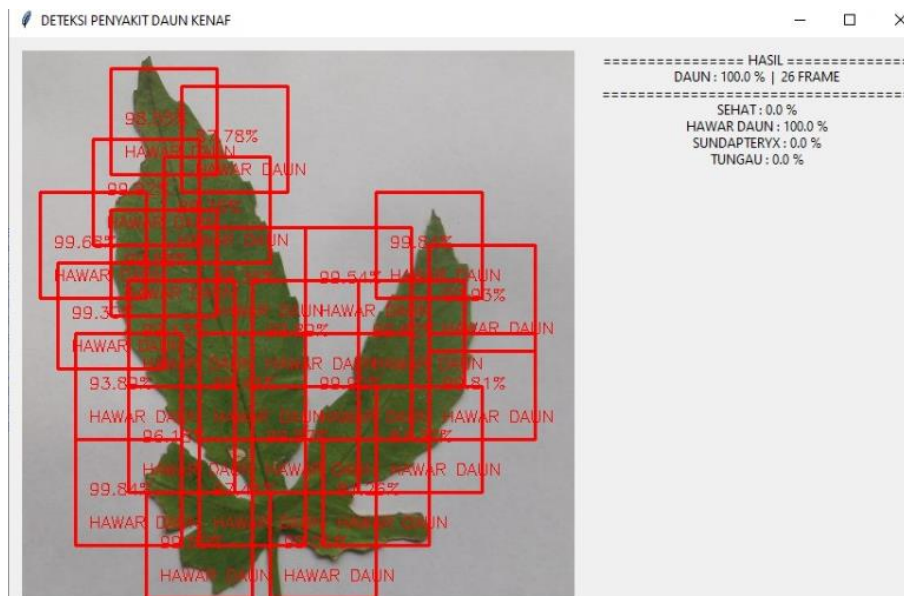


Fig. 16. Identification system detection with correct prediction

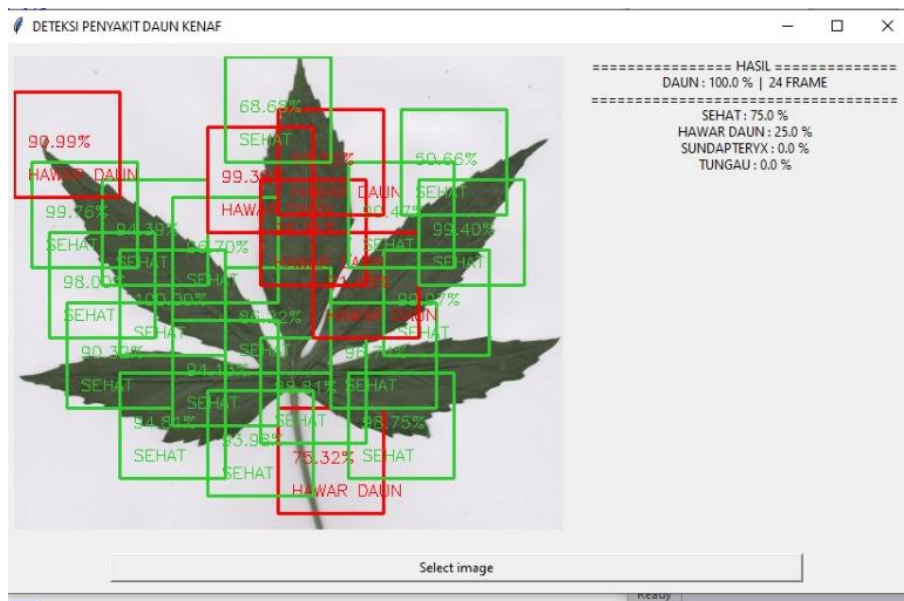


Fig. 17. System detection in healthy plant categories

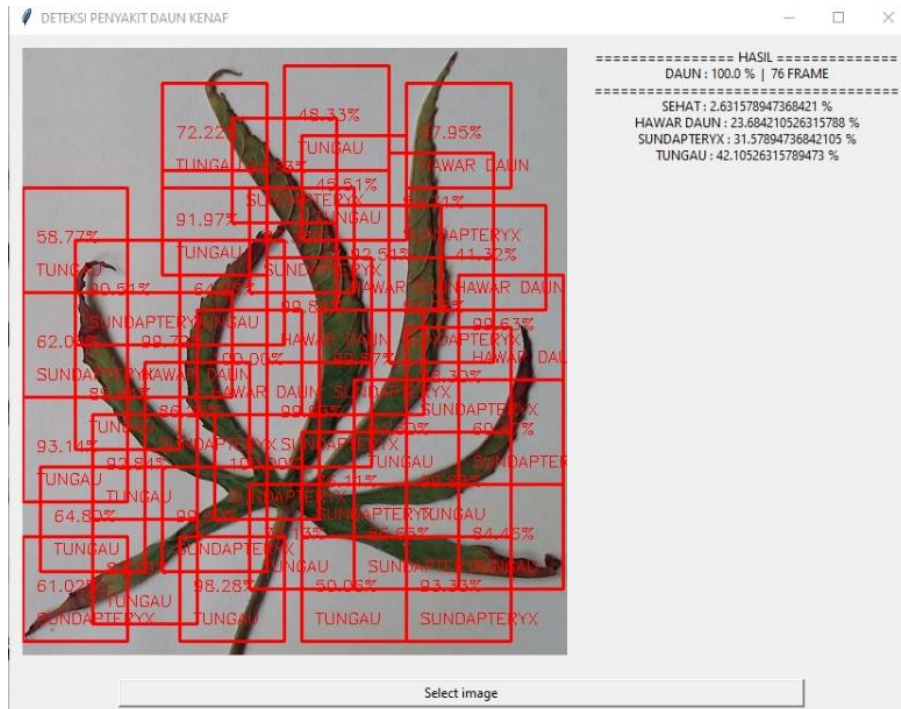


Fig. 18. The detection system has a prediction error

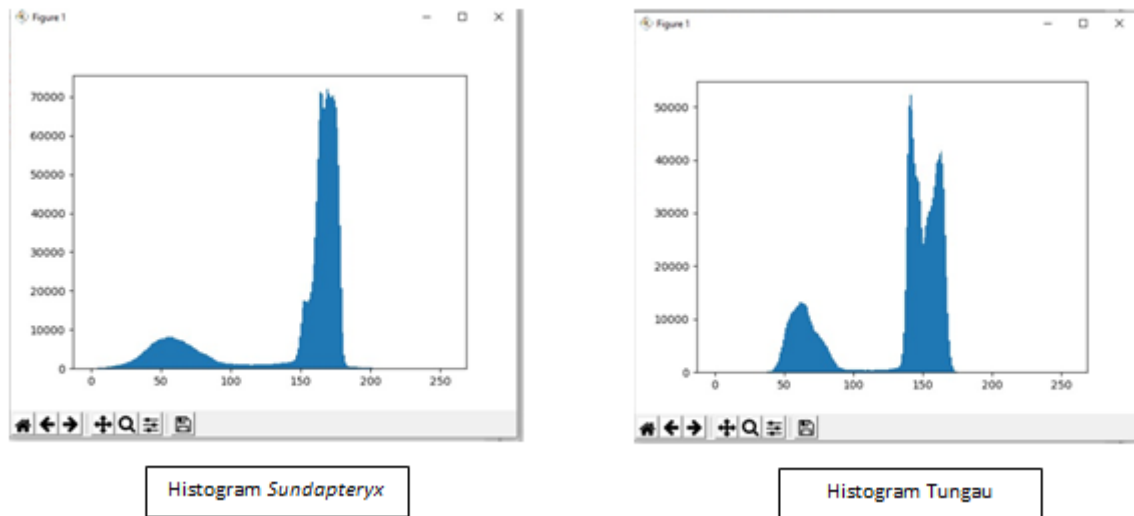


Fig. 19. Image histogram in the category of sundapteryx and mites

IV. Conclusion

In conclusion, the design of Kenaf plant disease & pest detection can be done by developing Image Recognition technology by coding using the Python programming language with open source Keras which is run on the TensorFlow machine learning platform. The architecture used in this study is the Convolutional Neural Network architecture: VGGNet 19 for leaf recognition and disease identification. The layer of the VGGNet19 architecture used for detection in this study includes the input layer, convolution layer, max pool layer, dense layer, dropout layer, and the addition of a flattening layer in the section on disease identification. Input feature training is given so that the characteristics formed from each disease cluster become more significant, thereby facilitating the classification process. The input feature training is designed by calculating the 5×5 pixel matrix block from the data of an image and calculating the average value so that a new matrix is formed, which is

then reprocessed using the specified architecture. The results of the introductory training achieved an accuracy value of 57.56% for the first two recognition classes and 25.37% for the second 4 classes on the VGGNet19 architecture. Based on the architectural design that has been established in this study, from several leaf data tested, namely 200 leaf data, 94 disease leaf images cannot be recognized correctly by the system and shown accurately at number 53%. Based on the results of the tests that have been carried out in this study, the convolutional neural network method can be recommended as a solution to an image-based recognition case. Some suggestions are given for further research, including adding more data so that the sample results can achieve maximum results. Then test more diverse types of architecture. Training on input features can be further expanded by providing more diverse block matrix training, such as 3×3, 8×8, 16×16 block matrices, etc. Add other methods such as the Adaptive fuzzy filter method or some other filtering method used to reduce noise or adjust the light entering the image to obtain more accurate training results through image data improvement.

Declarations

Author contribution

All authors contributed equally as the main contributor of this paper. All authors read and approved the final paper.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare no known conflict of financial interest or personal relationships that could have appeared to influence the work reported in this paper.

Additional information

Reprints and permission information is available at <http://journal2.um.ac.id/index.php/keds>.

Publisher's Note: Department of Electrical Engineering - Universitas Negeri Malang remains neutral with regard to jurisdictional claims and institutional affiliations.

References

- [1] Miyagawa and Tranggono, "Kebutuhan Serat Kenaf Sebagai Bahan Baku Industri PT TBINA," in *Seminar Nasional Serat Alam Serat Alam Inovasi Teknologi Serat Alam Mendukung Agroindustri yang Berkelanjutan*, 2015, pp. 54–59
- [2] E. Alpaydin, *Introduction to Machine Learning*. London: The MIT Press, 2004.
- [3] T. H. Saragih, W. F. Mahmudy, A. L. Abadi, D. M. N. Fajri, and Y. P. Anggodo, "Jatropha Curcas Disease Identification With Extreme Learning Machine," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 12, no. 2, 2018, doi: 10.11591/ijeecs.v12.i2.pp883-888.
- [4] M. Hassaballah and A. I. Awad, *Deep Learning in Computer Vision*, 1st edition. CRC Press, 2020.
- [5] J. Akbar, M. Shahzad, M. I. Malik, A. Ul-hasan, and F. Shafait, "Runway Detection and Localization in Aerial Images Using Deep Learning," *2019 Digit. Image Comput. Tech. Appl.*, pp. 1–8, 2019.
- [6] F. Ertam, "Deep learning based text classification with Web Scraping methods," in *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, 2018, pp. 1–4, doi: 10.1109/IDAP.2018.8620790.
- [7] S. T. Kebir and S. Mekaoui, "An Efficient Methodology of Brain Abnormalities Detection using CNN Deep Learning Network," in *2018 International Conference on Applied Smart Systems (ICASS)*, 2018, pp. 1–5, doi: 10.1109/ICASS.2018.8652054.
- [8] M. D. Radu, I. M. Costea, and V. A. Stan, "Automatic Traffic Sign Recognition Artificial Inteligence - Deep Learning Algorithm," 2020, doi: 10.1109/ECAI50035.2020.9223186.
- [9] P. Nepal, "VGGNet Architecture Explained," 2020. <https://medium.com/analytics-vidhya/vggnet-architecture-explained-e5c7318aa5b6>.
- [10] D. C. Khrisne and I. M. A. Suyadnya, "Indonesian Herbs and Spices Recognition using Smaller VGGNet-like Network," in *2018 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, Oct. 2018, pp. 221–224, doi: 10.1109/ICSGTEIS.2018.8709135.
- [11] S. V. Militante and B. D. Gerardo, "Detecting Sugarcane Diseases through Adaptive Deep Learning Models of Convolutional Neural Network," *2019 IEEE 6th Int. Conf. Eng. Technol. Appl. Sci.*, pp. 1–5, 2019, doi: 10.1109/ICETAS48360.2019.9117332.
- [12] D. M. N. Fajri, W. F. Mahmudy, and T. Yulianti, "Detection of Disease and Pest of Kenaf Plant using Convolutional Neural Network," *J. Inf. Technol. Comput. Sci.*, vol. 6, no. 1, p. 18, Apr. 2021, doi: 10.25126/jitecs.202161195.
- [13] T. Yulianti and Supriyono, "Penyakit Tanaman Kenaf dan Pengendaliannya," in *Monograf Balittas: Kenaf (Hibiscus cannabinus L.)*, 2009, p. 107.
- [14] P. Ranali, "A survey of Hemp Pest and Disease," in *Advances in Hemp Research*, P. Ranalli, Ed. Boca Raton, London,

New York: CRC Press Taylor & Francis Group, 1999, pp. 109–122.

- [15] J. K. Gill, “Automatic Log Analysis using Deep Learning and AI,” 2018.
- [16] Z. Monge, “Does Deep Learning Really Require ‘Big Data’? — No!,” *Medium Towards Data Science*, 2018.
- [17] Y. Le Cun et al., *Handwritten Digit Recognition with a Back-propagation Network*. San Francisco: Morgan Kaufmann Publishers Inc., 1990.
- [18] R. H. R. Hannloser, R. Sarpeshkar, M. A. Mahowald, R. Douglas, and H. S. Seung, “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit,” *Nature*, vol. 405, pp. 947–951, 2000, doi: <https://doi.org/10.1038/35016072>.
- [19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Network from Overfitting,” *J. Mach. Learn. Res.* 15, pp. 1929–1958, 2014.
- [20] D. Stathakis, “How many hidden layers and nodes?,” *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, 2009, doi: [10.1080/01431160802549278](https://doi.org/10.1080/01431160802549278).