Classification of Garlic Land Based on Growth Phase using Convolutional Neural Network

Durrotul Mukhibah, Imas Sukaesih Sitanggang, Annisa
Department of Computer Science,
IPB University, Bogor, Indonesia

***Abstract*—Indonesian Government needs to monitor the realization of garlic land with production plans in several production areas at growth season. A previous study, which used Sentinel-1A satellite imagery and Convolutional Neural Networks to classify garlic land, needed more information on growth phases. The study aims to address that limitation by creating a garlic land classification model based on the growth phase using Convolutional Neural Networks. The dataset comprises 446 preprocessed Sentinel-2 images cross-referenced with drone ground truth data. The model used both VGG16 and VGG19 architectures. Hyperparameter tuning was applied to obtain optimal values. After evaluating three scenarios (VGG16 base model, modified VGG16, and modified VGG19), the best model was obtained from the modified VGG19, which had an accuracy rate of 81.81% and a loss function of 0.71. The study successfully classified garlic land based on growth phase, with a precision rate of 0.43 for initial growth and vegetation classes, and 0.22 for the harvest class. The study offers an alternative to monitoring garlic production throughout growth phases with satellite imagery and deep learning.**

***Keywords*—Convolutional neural network; garlic; growth phase; horticulture; land classification; Sentinel-2; VGG**

I. Introduction

Garlic is a highly valued national food commodity in Indonesia, with distinctive characteristics and great market potential. According to the Central Bureau of Statistics, the market demand for garlic necessitates imports. Between 2019 and 2020, imports increased by 9.37% (US$ 51.29 million), while production decreased by 7.89% (7.02 thousand tons). Sembalun District in West Nusa Tenggara is a major garlic production hub, covering a harvest area of 2.47 thousand hectares and contributing 30.08% to the national garlic production [1].

The Indonesian government has implemented a policy to expand garlic cultivation to achieve self-sufficiency in garlic production. A key component of this policy involves the development of technology for monitoring garlic planting. The objective is to efficiently and inexpensively evaluate the suitability of land for garlic cultivation across different areas. Remote sensing has emerged as a widely adopted technology that enables large-scale mapping of agricultural landscapes at low cost and nearly in real-time [2].

There have been previous studies that utilized remote sensing to monitor garlic lands in the same area. Study [3] used remote sensing technology from Sentinel-1A satellites, employing C5.0 decision tree and Convolutional Neural Networks (CNN) to classify garlic/non-garlic land. The accuracy result shown that CNN accuracy is higher than decision tree. However, Sentinel-1A uses backscatter that can only reach land at the surface level, does not provide multi-temporal images, and the model has not applied growth phases. Additionally, the self-defined architecture (custom) used in this study is not a common architecture. Studies [5] and [6] extracted Sentinel-2 satellite NDVI during the growth phase using Random Forest (RF) and the Support Vector Machine (SVM) respectively. Both studies have shown that Sentinel-2 has great potential for use in garlic land classification based on growth phase, although its accuracy is currently low, at under 70%. The use of Machine Learning (ML) requires feature extraction and cannot process images as a data input and output directly.

This study aims to develop a land classification model for garlic using the CNN algorithm based on growth phase, building on the promising application of processing land classification by the CNN. Different with previous study [3], the model will take into account the growth phase of the garlic, which is important for district agencies to monitor the suitability of garlic cultivation during the planting season.

II. Literature Review

The European Space Agency (ESA) launched Sentinel-2, an open-access remote-sensing satellite that covers an area of up to 290 km. The satellite was developed with the primary mission of providing high-resolution satellite data for land cover and use, climate change and disaster monitoring. Sentinel-2 has a Multispectral Instrument (MSI) satellite with a band number of 13, a revisit time of five days and medium and high spatial resolution [4]. Previous studies have utilized Sentinel-2 imagery data for Land Cover and Land Use Classification (LCLU) in another area. According to study [5], Sentinel-2 has great potential to make early-season mapping of various types of winter crops, such as garlic and canola. In [6], Sentinel-2 is used to classify garlic fields using RF based on growth phase. In [7], the 10m multi-temporal red edge bands are the primary features of Sentinel-2 satellite data that are appropriate for analyzing LCLU.

The CNN has a potential algorithm to process land classification. Previous studies have implemented Deep Learning (DL) for LCLU in other area. CNN is a flexible Deep Learning (DL) algorithm that is commonly used for image recognition. It recognizes objects dynamically from various positions and shapes using a pixel-based approach [8]. According to study [9], DL algorithms can outperform ML for

processing text, image, video, voice, and speech data. DL can also automatically extract features based on the architecture. In [10], CNN outperforms other models in agricultural classification using Sentinel-2. In [11], CNN has advantages, such as weight-sharing features and simultaneous training for layer classification and feature extraction, which leads to a more stable and reliable model. The VGG16 and VGG19 architectures are considered the most successful CNN architectures due to the simplicity of network architecture. In [12], DL was implemented on Sentinel-2 images using U-Net architecture. In [13], there was a comparison between a Fully Convolutional Neural Network (FCN) with LSTM using a modified VGG19 encoder. In [14] adopted transfer learning to detect cracks using VGG16, ResNet18, DenseNet161, and AlexNet with pre-trained weights.

III. METHOD

The study was conducted in six steps: data collection, data preprocessing, data partition, hyperparameter tuning, CNN classification, model evaluation, and model comparison. Fig. 1 shows the steps of the study.

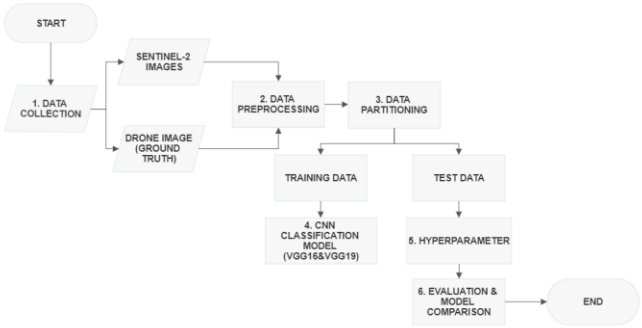

Fig. 1. Steps of study.

A. Data Collection

The study area is a major garlic production on the slopes of Mount Rinjani, Sembalun district, West Nusa Tenggara Province, Indonesia. Fig. 2 shows Sembalun district that is in 8°23 25.9"–8°22 06.4"S dan 116°31 32.9"–116°33 14.2".

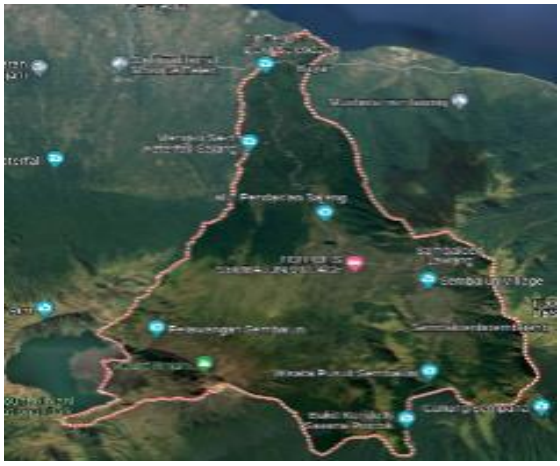

Fig. 2. Map of Sembalun District (source from Google Earth).

This study uses two data sources: drone images as ground truth and Sentinel-2 images. Drone images with a size of 43569 m × 36307 m were collected in the study [11] from 17 to 20 June 2021. Sentinel-2 data are free downloaded with a cloud cover of 2.64% at Copernicus (https://scihub.copernicus.eu/) for 1 July, 31 July, and 30 August 2021. The downloaded Sentinel imagery of the Sembalun at the L1C level was saved to a size of 10980 m × 10980 m. Sembalun garlic has a 105–110 days growth period [15]. The classes of growth phase are defined in Table I.

TABLE I. CLASSES OF GARLIC GROWTH PHASE

Class	Characteristic	Data sources
Initial growth phase	phase with soil and mulch still visible, more than 25% of soil is visible	Sentinel-2 1 July, 2021
Vegetative phase	phase with green dominant characteristics, the soil is no longer visible or less than 25% of soil is still visible (two months after the initial growth phase).	Sentinel-2 1 July, 2021
Harvest phase	phase with leaves drying out by more than 30% (two months after the vegetative phase	Sentinel-2 31 July and 30 August, 2021

During the initial growth and vegetation phase, Sentinel-2 data from drone images is available every 10-14 days. For the harvest period, there are two timeframes: July 31st and August 30th. During the first period, drone data is classified as vegetation and the Sentinel-2 data is taken one month or more after the drone vegetation phase. In the second period, if drone data is classified as early growth, then the Sentinel-2 data is taken two months or more after the early growth phase of the drone. This time difference aligns with the definition in Table I.

Preprocessing was performed in QGIS software using a semi-automatic classification plugin [21] to create imagery for CNN data sources. The first stage of data preprocessing is the atmospheric correction of Sentinel-2, which is required to eliminate the effects of scattering and absorption from the atmosphere to obtain surface reflectance characteristics [16]. The next step is band composite which uses RGB band composite [5]. In Sentinel-2, the RGB band composite is arranged by three of a spatial resolution of 10 m bands: four (red), three (green), and two (blue) [17]. The main characteristics of Sentinel-2 data suitable for land cover can be obtained from bands at a multitemporal resolution of 10 m [7].

A new raster of band composite was then resized to 209 m × 172 m to fit the drone images. In parallel, labeling for drones for class label initial growth and vegetative is done manually in referring to class label definition. Fig. 3 shows the labeling of drone images for initial growth and vegetation phases. However, harvest images in the drone were not available. Fig. 4 shows the labeling of Sentinel-2 imagery on 1st July using an overlay with labeled drone imagery.

Fig. 3. Labeling of drone images.

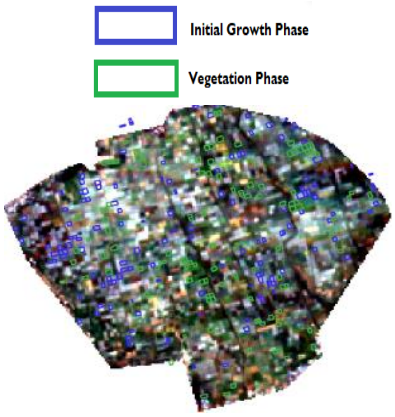

Fig. 4. Labeling Sentinel-2A 1 July images.

After labeling all Sentinel imagery, the last step is generating Tiff files with size 2 m × 2 m. Fig. 5, Fig. 6 and Fig. 7 illustrate how to generate the Tiff files for the initial growth, vegetation and harvest respectively.

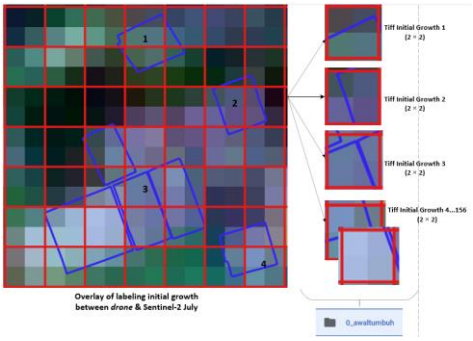

Fig. 5. Generating Tiff files of the initial growth phase.

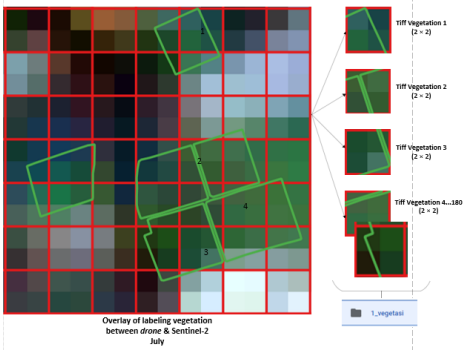

Fig. 6. Generating Tiff files of vegetative phase.

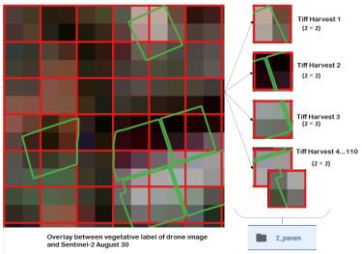

Fig. 7. Generating Tiff files of harvest phase.

The preprocessing results were tiff files of 156 initial growth, 180 vegetative, and 110 harvest phases. The files were then grouped based on the folder classification according to the growth phase. Classification folders were created as labeling references for CNN implementation.

B. Dataset Partition

The dataset was divided into training and test datasets randomly using random seeds in Python. A total of 446 images from the preprocessing results were split into training data of 402 (90%) and test data of 44 (10%). This partition consideration is enough for small dataset images.

C. Hyperparameter Tuning

Hyperparameter tuning is a step to modify the values of hyperparameters in CNN models to obtain optimal values. The hyperparameters include the number of layers, network size, epoch, batch size, and learning rate [18]. In the case of the number of layers, there was only the addition of two dense layers with activation function ReLu to VGG16 and VGG19 base models; for another tuning, by modifying some values manually and random search.

D. CNN Classification Model

In this step, a classification model of garlic land was developed based on growth phases using Keras library in Google Collaboratory GPU. The method follows the methodology of the study [3] with some modifications, detailed in Table II.

TABLE II. THE DIFFERENCE BETWEEN THE MODEL CREATION OF THE PREVIOUS STUDY AND THE CURRENT STUDY

Items	Previous Study [3]	Current study
Class	Binary (garlic/non-garlic)	Categorical (initial growth, vegetative, harvest)
Data input	Sentinel – 1A size 5 m × 5 m	Sentinel - 2 size 2 m × 2 m
Architecture	Custom: Input - convolution 1 - convolution 2 - flatten - fully connected 1- fully connected 2 - output	VGG16 & VGG19: input - convolution 1 - convolution 2 - convolution 3 - convolution 4 - convolution 5 - convolution 6 (VGG19) - fully connected 1 - (fully connected modification) - output
Pretrained	N/A	Imagenet
Data Augmentation	N/A	Zoom [0.0, 1.5]
Hyper parameters	epochs, batch size, and momentum	number of layers, epochs with early stopper, batch size, learning rate, and input size.

E. CNN Model Evaluation

The stages of model creation in the study are model architecture definition, pre-trained, data generator, training process, and model validation. The scenario is divided into three parts: Scenario A used VGG16 with a base model architecture, Scenario B used VGG16 with modification, and Scenario C used VGG19 with modification. The modification is the addition of two hidden layers of Dense with the ReLu activation function. All scenarios are applied to the same hyperparameters. Table III shows hyperparameters in scenarios A, B, and C.

The classification model performance is measured by confusion matrix, recall, and precision. The confusion matrix has four variables: TP (True Positive) represents actual data that is correct and predicted correctly, TN (True Negative) represents actual data that is incorrect and predicted incorrectly, FP (False Positive) represents actual data that is incorrect but predicted correctly, and FN (False Negative) represents actual data that is correct but predicted incorrectly. Accuracy (Acc), precision (Pr), and recall (Re) are formulated in Equations 1, 2, and 3 [19].

TABLE III. HYPERPARAMETERS IN SCENARIO A, B, AND C

Hyper parameters	Scenario A	Scenario B	Scenario C
Architecture	VGG16 default	Modified VGG16	Modified VGG19 and data augmentations
Input size network (width × height)	[64 × 64, 128 × 128, 224 × 224, 256 × 256]	[64 × 64, 128 × 128, 224 × 224, 256 × 256]	[64 × 64, 128 × 128, 224 × 224, 256 × 256]
Batch	[16, 32, 64, 128]	[16, 32, 64, 128]	[16, 32, 64, 128]
Optimizer	Adam	Adam	Adam
Epoch	300	300	300
Learning Rate	[0.0001, 0.001, 0.01]	[0.0001, 0.001, 0.01]	[0.0001, 0.001, 0.01]
Activation Function	ReLu	ReLu	ReLu

$$\text{Acc} = \frac{(N_{TP}+N_{TN})}{(N_{FP}+N_{FN}+N_{TP}+N_{TN})} \quad (1)$$

$$\text{Pr} = \frac{N_{TP}}{(N_{TP}+N_{FP})} \quad (2)$$

$$Re = \frac{N_{TP}}{(N_{TP}+N_{FN})} \quad (3)$$

$$H(p,y) = -\sum_i^n y_i \log(p_i), i \in [1,N] \quad (4)$$

In addition to these matrices, a loss function is a performance measure for models dedicated to CNN classification. The loss function is formulated in Equation 4 [11]. Equation 4 is used for the loss function cross-entropy with the softmax function. In layer output, CNN can calculate the prediction error generated by the CNN model through the training data using some loss function. The Loss Function uses two parameters to calculate the error, the first parameter is the estimated output of the CNN model (also called prediction), and the second is the actual output (also known as the label).

F. Development Environment

The development environment of the study is a notebook that was running on Windows 11. The hardware specifications are Intel(R) Core (TM) i7–8250U CPU @ 1.60GHz 1.80 GHz RAM 20 GB, SSD 500GB. The Software specifications are QGIS version 3.22.11-Białowieża, a google Collab Pro with GPU (https://colab.research.google.com/, was last accessed on 12 January 2023), including Keras modules for building model.

IV. RESULT AND DISCUSSIONS

All Tiff files from pre-processing were mounted to Google Drive. Classification folders are also created in google drive. Fig. 8 illustrates folder arrangement for folder classifications.

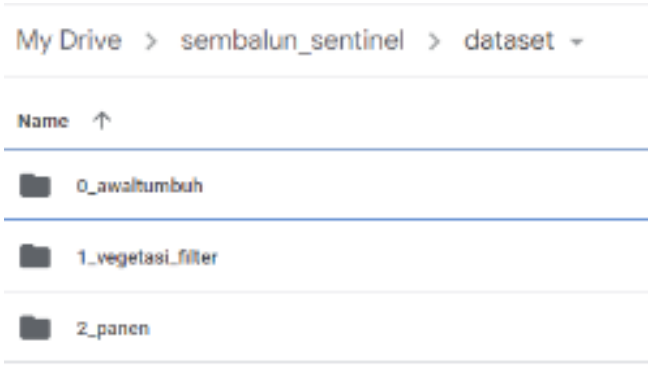

Fig. 8. Classification folders.

The model with VGG16 and VGG19 base model was loaded from Keras library using parameters network size 224 × 224 (default VGG) and pre-trained with ImageNet. The Pre-trained model, which contains 80,134,624 data, was downloaded from Keras storage. The modification of layers is assembled by adding two hidden layers dense (2048) and ReLu activation on base models VGG16 and VGG19. The addition of layers increased the total parameters processed. The full parameters of VGG16 base model before and after adding layers are 14,789,955 and 70,299,459, respectively. The total number of parameters of the VGG19 base model before and after adding layers is 20,024,384 and 75,609,155, respectively. Fig. 9 illustrates the difference between the VGG16 base model before and after adding layers.

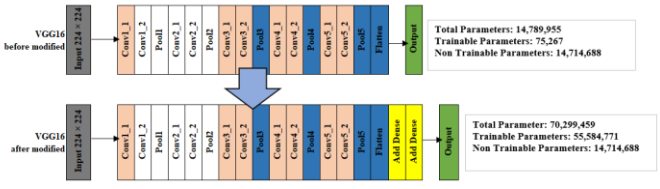

Fig. 9. VGG16 model architecture before and after adding hidden layers.

Training and testing steps used Image Data Generator module from Keras library by applying the Zoom Range =1.5 for data augmentations. The flow from the directory module processes training and test data according to the batch size and network input and then automatically labels the growth phase class. The process of labeling input images identified the classification folders. Hyperparameters use

optimizer = adam, loss = categorical_crossentropy, and metrics = categorical_accuracy. The structure of a new layer with the ReLU activation function observes the default convolution of VGG. The model was trained with an epoch of 300 and applied an early stopper during the training process. As a result of the training process, each image already has a categorical type of class, namely the initial classes of growth (0), vegetation (1), and harvest (2). The training results on scenario models A, B, and C are shown in Fig. 10, Fig. 11, and Fig. 12, respectively.

Input size network (width×height)	Hyper-parameters				Accuracy (%)	Loss
	Batch	Learning Rate	Optimizer	Epoch(max)		
224 × 224	64	**0.001**	Adam	76	83.0	0.46
224 × 224	**32**	0.001	Adam	97	82.5	0.44
256 × 256	32	0.001	Adam	56	84.0	0.41

Fig. 10. Training results of scenario A using the VGG16 base model.

Input size network (width×height)	Hyper-parameters				Accuracy (%)	Loss
	Batch	Learning Rate	Optimizer	Epoch(max)		
224 × 224	64	**0.001**	Adam	76	83.0	0.46
224 × 224	**32**	0.001	Adam	97	82.5	0.44
256 × 256	32	0.001	Adam	56	84.0	0.41

Fig. 11. Training results of the scenario B using VGG16 modifications.

Input size network (width×height)	Hyper-parameters				Accuracy (%)	Loss
	Batch	Learning Rate	Optimizer	Epoch(max)		
224 × 224	64	**0.001**	Adam	92	77.11	0.54
224 × 224	**32**	0.001	Adam	97	81.0	0.49
64 × 64 (Zoom=0)	32	0.001	Adam	145	87.81	0.29
64 × 64 (Zoom =1.5)	32	0.001	Adam	86	80.59	0.50

Fig. 12. Training results of the scenario C using VGG19 modifications.

The best model was acquired from scenario C with network input = 64 × 64, batch = 32, learning rate = 0.001, zoom = 0.0, and optimizer = Adam. The training time required is 111 seconds. CNN models were evaluated using test data. The best model results an accuracy of 81.81% and loss function of 0.71. The best model has corrected predictions of two, nine, and six for initial growth, vegetative, and harvest, respectively. It predicted initial growth as nine for vegetation, which is the correct prediction. The prediction results are shown in a confusion matrix in Table IV. The precision call, recall values for the corrected predictions are shown in Table V.

TABLE IV. CONFUSION MATRIX

Class	Initial growth	Vegetation	Harvest
Initial growth	2	9	4
Vegetative	5	9	4
Harvest	2	3	6

TABLE V. PRECISION AND RECALL

Class	Precision	Recall
Initial growth	0.22	0.13
Vegetative	0.43	0.50
Harvest	0.43	0.55

The addition of the growth phase is tested to the same architecture from the previous study [3]. An accuracy result of 76.40% and a loss function of 0.65 with input size = 128 × 128, batch = 64, and learning rate = 0.0001.

Based on the three tested scenarios, scenario C's best results were obtained using the VGG19 model with the addition of hidden layers. The loss function is fundamental in selecting the best scenario in CNN; a lower loss function is considered better. The best model has processed 28,423,235 parameters with a loss function of 0.71 and an accuracy of 81.81% on test data. The best model used hyperparameters such as input network = 64 × 64, batch = 32, learning rate = 0.001, zoom = 0.0, optimizer = adam, and epoch = 145. The difference in loss between scenario A and scenario B in the VGG16 architecture demonstrates the impact of architectural changes. Scenario B, which includes additional hidden layers, decreased losses by 0.09 and an accuracy increase of 2.81 compared to Scenario A. This trend continued in Scenario C, where modifying the VGG19 architecture decreased loss by 0.03 and 1.00 increase in accuracy compared to Scenario B. The hidden layer is the most important layer in the CNN architecture, as it builds several other layers based on user requirements [20].

The number of parameters processed depends on the network input. A higher input network leads to an increase in the number of processed parameters. The batch and learning rates are interrelated and significantly affect the model's convergence rate and the number of epochs. A larger batch decreases epochs, while a greater learning rate increases the number of epochs. In VGG19 with transfer learning, the hyperparameters learning rate and epochs provide the best classification network [21]. The use of zoom data augmentation in both training and test data has not increased accuracy, which contrasts with [22]. Another consideration of the best model is observed through its convergent level, which indicates its stability. The convergent levels of scenario models A, B, and C on the test data are presented in Fig. 13, Fig. 14, and Fig. 15, respectively. Those graphs demonstrate that scenario C has a higher convergent rate than the other scenarios.

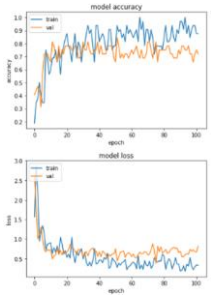

Fig. 13. Accuracy and loss scenario A accuracy = 72.72%, loss = 0.60.

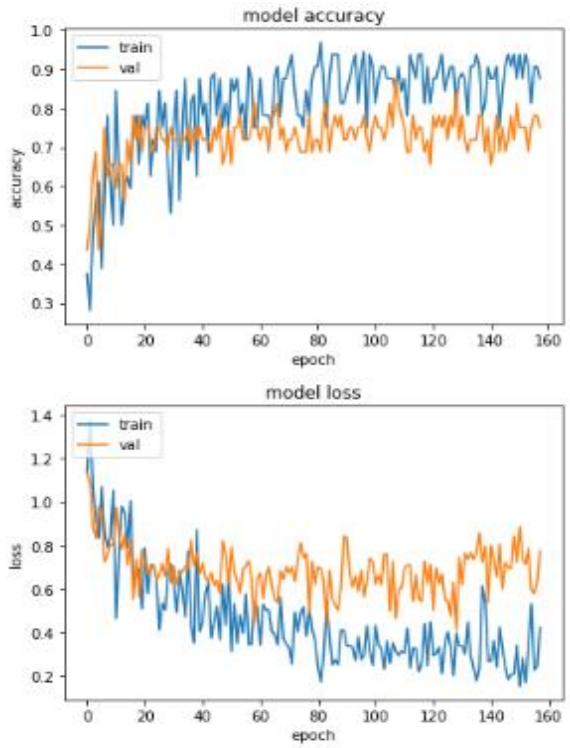

Fig. 14. Accuracy and loss scenario B accuracy = 77.27%, loss = 0.63.

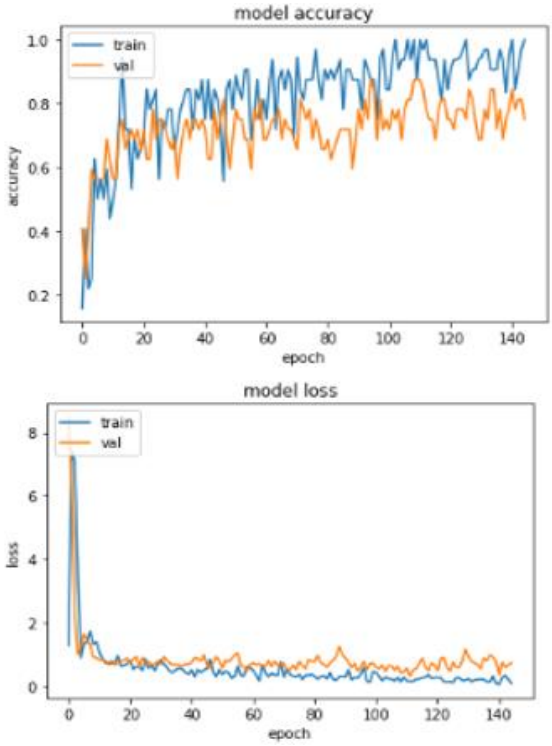

Fig. 15. Accuracy and loss scenario C accuracy = 81.81%, loss = 0.71.

The best model can accurately classify vegetation and harvest phases with a precision of 0.43, but it has not yet been able to accurately detect the initial growth phase, with a precision of 0.22. The initial growth phase is mainly classified as vegetation.

In contrast to the previous study [3], the best model of the study did not result in better accuracy or loss function. The accuracy decreased by 4.65 or 5.35% from the original 86.46%. The loss function value increased by 0.22 or 44.89% from the original 0.49. These differences between the studies can be attributed mainly to the increase of number of classification classes, which changed from binary to categorical, as well as to the utilization of a different model architecture. The study also evaluated the architecture from the previous study using the newer categorical class follow the study, resulting in an accuracy of 76.40% and a loss of 0.65, a lower than previous. This evaluation demonstrates that the same architecture with a different class result in a decreased accuracy. However, further testing and training with other scenarios are needed to achieve a suitable model for garlic land classification.

V. Conclusion

The study has successfully developed a model to classify garlic lands based on growth phases using the CNN algorithm and Sentinel 2 imagery. The best model achieved an accuracy of 81.81% and a loss function of 0.71. The best model was obtained from modified VGG19 architecture with pre-trained weights, no data augmentation, and hyperparameters: input network, batch size, learning rate, optimizer, and epoch. The best model can classify the vegetation and harvest phases, but the initial growth phases need to be better classified, with most initial growth classes being classified as vegetation.

The study has shown that CNN is a promising method for processing multitemporal imagery in seasonal agriculture land cover. For future work, the Sentinel-2 images can be resized in various sizes to ensure a precision size of network input of CNN. The Sentinel-2 band composite should also consider using NIR and SWIR instead of RGB because some studies used NIR and SWIR, which a good band for agriculture. Additionally, it may be beneficial to consider labeling for other growth phases and land types.

As garlic is a seasonal plant, it is possible to plant another crop in an area that previously used to grow garlic, which could potentially be labeled as a garlic vegetative phase. Furthermore, to improve CNN modeling, dataset partition, hyperparameters such as architecture, learning rate, loss function, and optimizer should be fine-tuned before considering additional data input. Stakeholders and researchers can use this study as an alternative of monitoring garlic land along growth season through satellite imagery and deep learning to support import policy.

Acknowledgment

The authors wish to thank the Institutional Research Grant of Agro-Maritime, IPB University, Contract No. 7823/IT3.L1/PT.01.03/P/B/2021 for financial support.

References

[1] BPS, “Statistik hortikultura 2020 (in Bahasa),” Badan Pusat Statistik, 2020. https://www.bps.go.id/publication/2021/06/07/daeb50a95e860581b20a2ec9/statistik-hortikultura-2020.html (accessed Sep. 17, 2021).

[2] D. Kpienbaareh et al., “Crop type and land cover mapping in northern malawi using the integration of sentinel-1, sentinel-2, and planetscope

satellite data," Remote Sens., vol. 13, no. 4, pp. 1–21, 2021, doi: 10.3390/rs13040700.

[3] R. I. Komaraasih, I. S. Sitanggang, and M. A. Agmalaro, "Sentinel-1A image classification for identification of garlic plants using decision tree and convolutional neural network," vol. 11, no. 4, pp. 1323–1332, 2022, doi: 10.11591/ijai.v11.i4.pp1323-1332.

[4] ESA, Sentinel-2 user handbook, vol. 48, no. 9. 2013.

[5] T. Haifeng, W. Yongjiu, C. Ting, L. Zhang, and Q. Yaochen, "Early season mapping of winter crops using Sentinel-2 optical imagery," MDPI, pp. 1–11, 2021, doi: 10.3390/rs13193822.

[6] Z. Chai, H. Zhang, X. Xu, and L. Zhang, "Garlic mapping for Sentinel-2 time series data using a random forest classifier," pp. 7224–7227, 2019, doi: 10.1109/igarss.2019.8900617.

[7] A. M. Abdi, "Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data," GIScience Remote Sens., vol. 57, no. 1, pp. 1–20, 2020, doi: 10.1080/15481603.2019.1650447.

[8] A. Vali, S. Comai, and M. Matteucci, "Deep learning for land use and land cover classification based on hyperspectral and multispectral earth observation data: A review," Remote Sens., vol. 12, no. 15, 2020, doi: 10.3390/RS12152495.

[9] J. Díaz-Ramírez, "Machine Learning and Deep Learning," Ingeniare, vol. 29, no. 2, pp. 182–183, 2021, doi: 10.4067/S0718-33052021000200180.

[10] A. M. Simón Sánchez, J. González-Piqueras, L. de la Ossa, and A. Calera, "Convolutional Neural Networks for Agricultural Land Use Classification from Sentinel-2 Image Time Series," Remote Sens., vol. 14, no. 21, 2022, doi: 10.3390/rs14215373.

[11] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," Intell. Syst. Ref. Libr., vol. 172, no. June, pp. 519–567, 2019, doi: 10.1007/978-3-030-32644-9_36.

[12] G. Singh, S. Singh, G. Sethi, and V. Sood, "Deep Learning in the Mapping of Agricultural Land Use Using Sentinel-2 Satellite Data," Geographies, vol. 2, no. 4, pp. 691–700, 2022, doi: 10.3390/geographies2040042.

[13] O. Sefrin and F. M. Riese, "Deep Learning for Land Cover Change Detection," 2021.

[14] M. M. Islam, M. B. Hossain, M. N. Akhtar, M. A. Moni, and K. F. Hasan, "CNN Based on Transfer Learning Models Using Data Augmentation and Transformation for Detection of Concrete Crack," Algorithms, vol. 15, no. 8, 2022, doi: 10.3390/a15080287.

[15] L. Sandrakirana, Ratih, "Panduan budidaya bawang putih," Balai Pengkajian Teknologi Pertanian Jawa Timur, 2018. https://jatim.litbang.pertanian.go.id/wp-content/uploads/2019/04/BAWANG-PUTIH-3.pdf (accessed Nov. 21, 2021).

[16] Khairunnisa, Annisa, and I. S. Sitanggang, "Application of Random Forest Algorithm on Sentinel-2A Imagery for Garlic Land Classification Based on Growing Phase in Sembalun," 2022 IEEE Int. Conf. Aerosp. Electron. Remote Sens. Technol. ICARES 2022 - Proc., no. 7823, 2022, doi: 10.1109/ICARES56907.2022.9993563.

[17] H. Tian, J. Pei, J. Huang, X. Li, J. Wang, and B. Zhou, "Garlic and Winter Wheat Identification Based on Active and Passive Satellite Imagery and the Google Earth Engine in Northern China."

[18] W. Zhu, J. Chen, D. Chen, and Y. Lin, "Evolutionary Convolutional Neural Networks Using ABC," no. February, 2019, doi: 10.1145/3318299.3318301.

[19] M. Kubat, An Introduction to Machine Learning. 2017.

[20] P. S. Rathore, N. Dadich, A. Jha, and D. Pradhan, "Effect of Learning Rate on Neural Network and Convolutional Neural Network," Int. J. Eng. Res. Technol., vol. 6, no. 17, pp. 1–8, 2018.

[21] T. H. Nguyen, T. N. Nguyen, and B. V. Ngo, "A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease," AgriEngineering, vol. 4, no. 4, pp. 871–887, 2022, doi: 10.3390/agriengineering4040056.

[22] R. Gharbia, N. E. M. Khalifa, and A. E. Hassanien, "Land Cover Classification Using Deep Convolutional Neural Networks," no. June, pp. 911–920, 2021, doi: 10.1007/978-3-030-71187-0_84.