Journal of Information Hiding and Multimedia Signal Processing ©2025 ISSN 2073-4212
Ubiquitous International Volume **16**, Number **2**, June **2025**

Microscopic Image Classification of Fungal Colonies using VGG Pre-trained Model

R Putri Ayu Pramesti[1,2,*], Muhamad Rodhi Supriyadi[1,2], Muhammad Reza Alfin[1,2], Josua Geovani Pinem[1,2], Aulia Haritsudin Karisma Muhammad Subekti[1,2], Umi Chasanah[1,2], Gilang Mantara Putra[1,2], Mukti Wibowo[1,2], Dewi Habsari Budiarti[1,2], Jemie Muliadi[1,2], Anto Satriyo Nugroho[1,2]

[1]Artificial Intelligence and Cyber Security Research Center
National Research and Innovation Agency of Indonesia
[2]Bandung, West Java, Indonesia
{rade032, muha242, muha163, josu001, auli008, umic001, gila003, mukt003, dewi037, jemi001, anto006}@brin.go.id

Danang Waluyo[3,4]

[3]Vaccine and Drug Research Center
National Research and Innovation Agency of Indonesia
[4]Bogor, West Java, Indonesia
dana003@brin.go.id

Received December 30, 2024, revised March 5, 2025, accepted March 6, 2025.

Abstract. *Deep Learning is a method that has been widely used to perform object recognition, including object recognition based on microscopic imagery. Even so, until now there have not been many studies that identify fungus based on microscopic images. Because fungi itself is a microorganism that lives in colonies, the identification process has its own difficulties. In this research, the classification of fungi from the genera Aspergillus, Cladosporium, Trichoderma, Fusarium, and Penicillium. The architecture used in this study was the VGG16 Pre-Trained Model. Several tests were carried out with two types of datasets, namely the imbalanced dataset and the balanced dataset using the downsampling technique. The experiments are conducted to compare the performance of the baseline architecture of VGG16 and the customized VGG16 where the size of the fully connected layers was reduced. The results indicate that the imbalanced dataset's accuracy with the customized VGG16 was the highest, which was 79.6.%. In this study, K-fold cross-validation succeeded in increasing accuracy to 82%.*
Keywords: Fungi classification, Deep-Learning, VGG16, pre-trained model, K-Fold cross-validation.

1. **Introduction.** Fungi is a type of microorganism that has an essential role in everyday life. People are more familiar with fungus as harmful microorganisms closely related to diseases. However, in reality, some types of fungus have benefits in various fields. For example, Aspergillus has several benefits, including a compound that produces anti-cancer and anti-fungal properties[1]. It also plays a role in the production of food enzymes[2]. Besides, another genus, namely Trichoderma, acts as a plant growth simulator[3] and biocontrol agent[4], while Penicillium is a genus known as an antibiotic[5]. The expert will identify the fungus type so they can use it for various needs. Experience and deep knowledge are needed to be able to identify the type of fungi. Artificial intelligence has been widely used to perform image-based object recognition processes quickly and efficiently.

A previous research[6] used classical machine learning methods to classify Aspergillus, Cladosporium, and Trichoderma genera. The accuracy obtained is still relatively low. An alternative that can be used is the Deep Learning method. The application of Deep Learning, which is part of artificial intelligence, can learn large and complex data by imitating the work of the human brain[7], becoming one of the most effective tools in object recognition. Deep Learning is considered a technique with good capabilities in performing feature extraction, accurate recognition, and high speed[8]. Deep Learning has been widely used for image-based object identification and detection, including microscopic images. Research[9] used deep Learning to help diagnose leukemia using microscopic blood cell images. Another research[10] used microscopic image datasets to diagnose cancer using deep learning methods. One of the Deep Learning architectures that can be used is VGGNet. VGGNet was developed by the Visual Geometry Group from the University of Oxford. VGGNet was the runner-up of ILSVRC (ImageNet Large Scale Visual Recognition Competition) in 2014[11]. This architecture identifies the effect of accuracy on the depth of the CNN network in image recognition. The main contribution of this architecture is, increasing the depth of the network using smaller (3×3) convolution filters. VGG 16 has 16 layers, 13 convolution layers, and 3 fully connected layers. Research[12] performs a comparison of several architectures to build a plankton classification model. The results show that the VGGNet architecture has the highest accuracy compared to other architectures such as Alexnet, Googlenet, InceptionV3, Resnet, and Densenet. Other research[13] that used CNN to classify leukocytes showed the results of VGGNet's accuracy were better than those of AlexNet. This research applies the VGG16 Deep Learning architecture, with the transfer learning method to perform feature extraction and image classification consisting of 5 fungus's genera (Aspergillus, Cladosporium, Trichoderma, Fusarium, and Penicillium). Training deep learning models, as is well known, requires significant computational resources. Many studies have been conducted to reduce the computational cost. Such as cutting the number of layers or reducing the filter size[14]. In this study, an experiment was conducted by reducing the size of the full-connected layer and seeing its effect on the training process and the resulting model. Thus, the expectation is that the resource load will be lighter and the resulting model size will be smaller so that it can help identify fungus more quickly and accurately with relatively low computational cost.

2. **Method.**

2.1. **Dataset.** The data used is microscopic image data from the BRIN Biotech Lab, which was taken in 2021 consisting of 5 genera, namely Aspergillus, Penicillium, Trichoderma, Cladosporium and Fusarium in JPG, TIF and TIFF formats. The details of the number of datasets available can be seen in Table 1. The total number of images is 3174 images. Each genus has a different amount of data. Aspergillus has 835 images, Cladosporium has 550 images, Fusarium has 373 images, Penicillium has 1159 images, and Trichoderma has 257 images. Each image has a high level of variation, including image magnification, colouring, colony density, and variations in fungal age. All images are RGB images with different image sizes. The examples of microscopic images of each genus can be seen in Figure 1.

Two types of dataset were used. The first dataset consists 50 image data as testing data, and the rest of the image dataset was used as training data. The second dataset used the down-sampling technique. The dataset undergoes down-sampling in order to address imbalances. 100 images data were taken in each class to be used as training data, then the remaining images in each class were used as data testing.

TABLE 1. Imbalanced dataset

Genus	Number of images	Size
Aspergillus	835	5.99 GB
Cladosporium	550	3.19 GB
Fusarium	373	2.80 GB
Penicillium	1159	7.08 GB
Trichoderma	257	1.48 GB

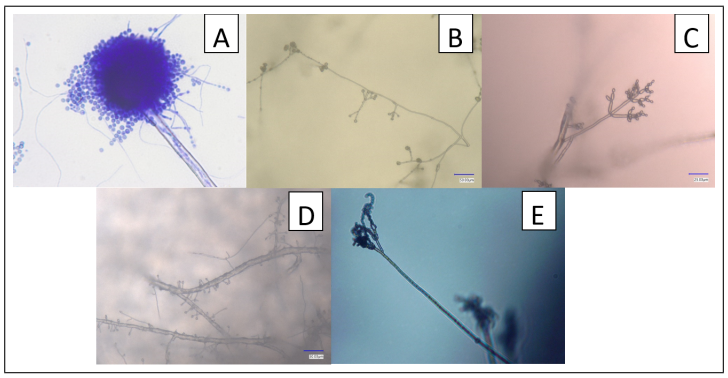

FIGURE 1. Fungus microscopic image examples

2.2. **Pre-Processing.** The fungus dataset images used have various sizes. Each image is then resized to a size of 244x244x3, since these input metrics were used to train VGG16 initially. After that, image augmentation is carried out with the variant process, this is done to prevent overfitting of the Deep Learning model[15]. Augmentation overcomes overfitting by minimizing the distance between the training and validation set[16]. Because deep learning work rely on large datasets, image augmentation is considerably crucial for overcoming the limitations of data samples, particularly image data-sets[17]. In all of experiments, training and validation data were randomly divided into the training set and validation set with 80:20 of proportion. The pre-processing stages illustration can be seen in Figure 2.

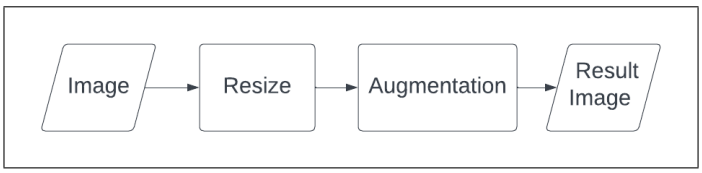

FIGURE 2. Pre-processing stages illustration

2.3. **Pre-Trained Model.** The pre-trained model is a solution for limited datasets and high computing devices. Pre-trained models are created by training many datasets with high variations to create a machine-learning algorithm that can capture unique information from an object. ImageNet is a dataset that is commonly used to create pre-trained models. ImageNet is considered superior to other datasets, especially because ImageNet has consistently won the Large Scale Visual Recognition Challenge (ILSVRC) since 2021. Several pre-trained ImageNet models can be used, including VGGNet, AlexNet, and ResNet. VGGNet was developed by Simonyan, Zisserman of the University of Oxford's Visual Geometry Group. Pre-trained VGGNet is built using 138 million parameter[18]

and produces Top-5 error rates of 7.3%. This pre-trained was then used in this study to form a new model with the fungic dataset. VGGNet is designed to reduce AlexNet's large kernel sizes of 1x1 and 5x5 and replace them with several kernels of 3x3 and pooling sizes of 2x2. VGGNet performs network depth enhancement which is useful for extracting complex features. Simonyan, Zisserman created a VGG architecture with several depths and found that the most significant results were obtained with layer 16 and 19 depths. VGG16 is VGGNet with layer 16 depth, consisting of 13 convolution layers and 3 fully connected layers[19]. VGG16 performs training on 224x224 size of images[20]. The number of channels of the convolution layer starts from 64, then increases and eventually becomes 512 channels. An architectural illustration of the VGG16 can be seen in Figure 3. In this research, the training process was carried out using the VGG16 architecture,

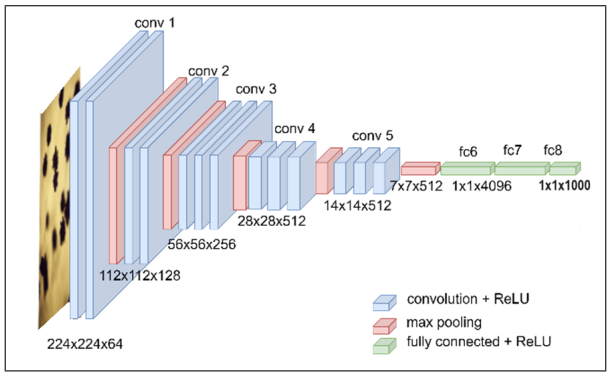

FIGURE 3. VGG16's architectural illustration

where the size of the fully connected layer was reduced to 100. These changes greatly minimize the number of trainable parameters. Therefore, the training process becomes faster, and the model size becomes smaller. The difference between the baseline VGG16 architecture and the customized VGG16 can be seen in the illustration in Figure 4.

2.4. **Transfer Learning.** Transfer learning is a fairly common method used to improve accuracy with limited datasets. Transfer learning utilizes the knowledge gained from previous assignments to be used in other tasks by making modifications to some of the last layers[21]. By studying previously acquired knowledge, transfer learning can perform tasks in different domains without having to do Learning from scratch which has implications for very large data requirements[22]. The basic idea underlying transfer learning is to assist a machine learning algorithm in achieving better performance in the area of interest by borrowing labelled data or knowledge collected from some related domains. This study uses ImageNet transfer learning which previously classified 1,000 ImageNet dataset categories of 14 million images[23]. Modifications are made to the fully connected layer and the number of classes. The number of classes is 5 classes which are the genera of the fungi being classified. The illustration of transfer learning architecture can be seen in Figure 5.

2.5. **Training.** At the training stage, four experiments were carried out on two types of dataset distribution. The four experiments are:

- The first experiment was carried out using a baseline architecture with an imbalanced dataset
- The second experiment was carried out with the customised VGG16 using an imbalanced dataset
- The third experiment was carried out with the customised VGG16 using a dataset

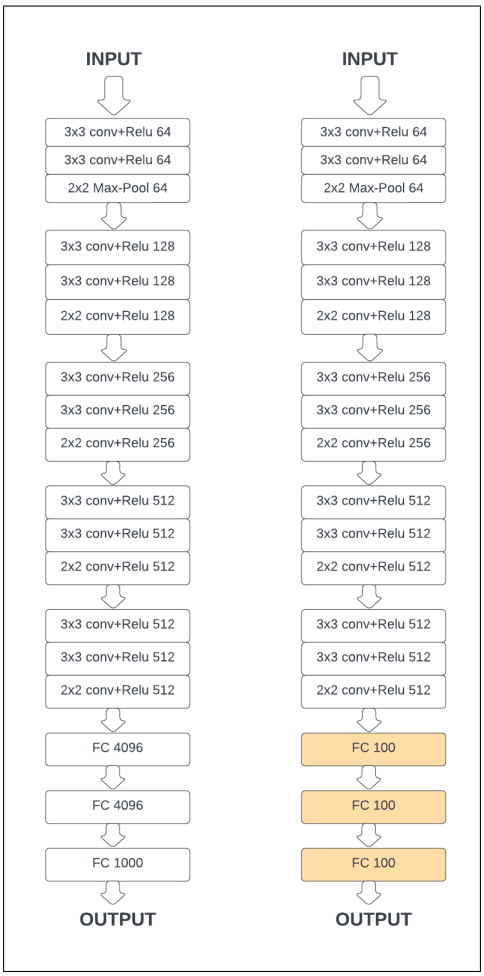

FIGURE 4. Baseline(left) and proposed architecture(right)

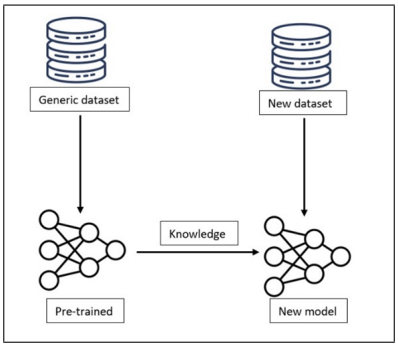

FIGURE 5. The illustration of transfer learning architecture

down-sampling technique

• The fourth experiment was carried out using imbalanced dataset with K-Fold cross validation to overcome overfitting problem.

Parameters used for all experiment during training can be seen in Table 2.

2.6. **Evaluation.** To see the performance of the model that has been made, accuracy, recall and precision calculations are performed. Accuracy is the percentage of all correctly identified samples to the total. Although accuracy is a highly obvious evaluation metric, it is not always reliable. As a result, additional indications must be assessed in order to

TABLE 2. Experiment's parameters

Parameter	Setting
Training:validation data ratio	8:2
Deep-learning architecture	VGG16
Cost function	Categorical
Epoch	50
Activation	Softmax
Drop out	0.2
Regularization	0.2
Learning rate	0.001

gauge the model's correctness. Recall is the ratio of true positive predictions compared to all data that really belongs to the class[24], while precision is the ratio of true positive predictions compared to all positive predicted results. Precision reflects of how many selected items are relevant, while recall reflect how many relevant items are selected[25]. The metrics of Recall and Precision can be seen in Figure 6, and the formula for calculating Recall (R) and Precision (P) can be seen in (eq1) and (eq2).

$$R = \frac{TP}{TP + FP} \tag{1}$$

$$P = \frac{TP}{TP + FN} \tag{2}$$

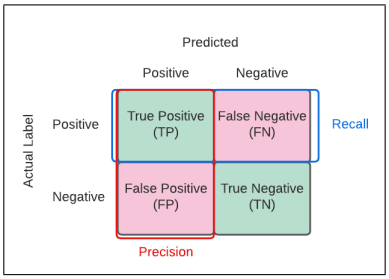

FIGURE 6. Metric of recall and precision

Where TP is the number of objects labelled as class A and correctly predicted, FN is an object of class A but predicted not to enter class A, and FP is an object of another class but predicted to enter class A. In addition to recall and precision, F1 Score is also calculated, which is a comparison of the weighted average precision and recall. How to calculate the F1 score can be seen in Equation (3).

3. **Result and Discussion.** The first and second models were generated from training with the baseline architecture and customised VGG16 using the imbalanced dataset. Testing is done using each of the 50 images for each class. The distribution of training and testing data for the imbalanced dataset can be seen in the Table 3. The two models are then used in the testing stage to see their performance. The performance of each model can be seen in the Table 4.

According to the comparison of the two result models, reducing the size of the fully connected layer speeds up the training process compared to the baseline architecture; additionally, the result model size is smaller, which is obviously preferable because when

TABLE 3. Imbalanced dataset

Genus	Training imagesl	Testing images	Total
Aspergillus	758	50	835
Cladosporium	500	50	550
Fusarium	323	50	375
Penicillium	1109	50	1159
Trichoderma	207	50	257

TABLE 4. Model's performance

Parameters	Baseline	Proposed
Running time (second)	7705	6215
Model size	540,541 KB	62,422KB
Accuracy	76.0%	79.6%

the model is implemented into an application, the computational load becomes lighter. Although the customised VGG16 model size is smaller, its accuracy remains marginally higher than the model produced by the baseline architecture.

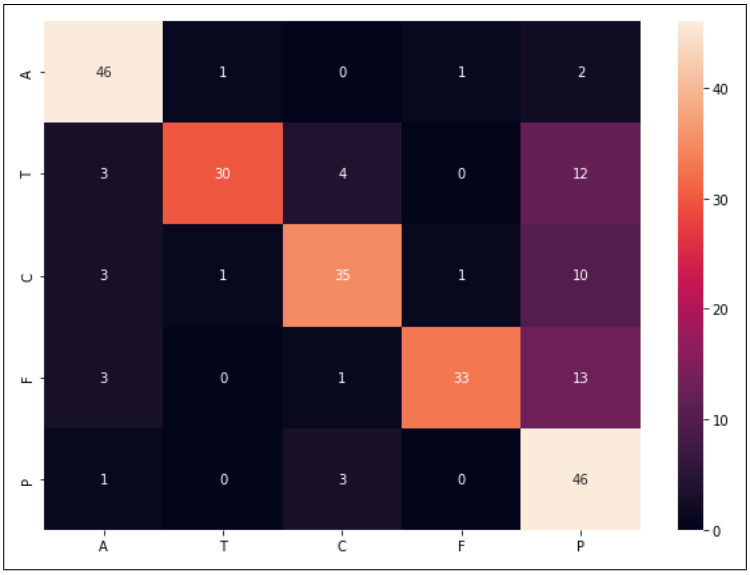

FIGURE 7. Confusion matrix of imbalanced dataset with baseline architecture testing

A confusion matrix, as shown in Figure 7, is derived for the first model, which was created using the baseline architecture and an imbalanced dataset. The correct classification results for Aspergillus and Penicillium got the highest numbers, while other classes, namely Cladosporium, Trichoderma, and Fusarium got numbers between 30-35. The confusion matrix shows that inaccurate prediction data tends to go to the Penicillium class. The second confusion matrix, as shown in Figure 8 is generated using the customised VGG16. In this model, the highest number of correct predictions was obtained by the aspergillus and fusarium classes. In the two confusion matrices produced, it can be seen that in both confusion matrices there is a bias for incorrect predictions tend to go to Penicillium class. Those two confusion matrix show that there is a bias for data to fall into specific classes. In this case, the data has a bias to fall into the Penicillium class; based

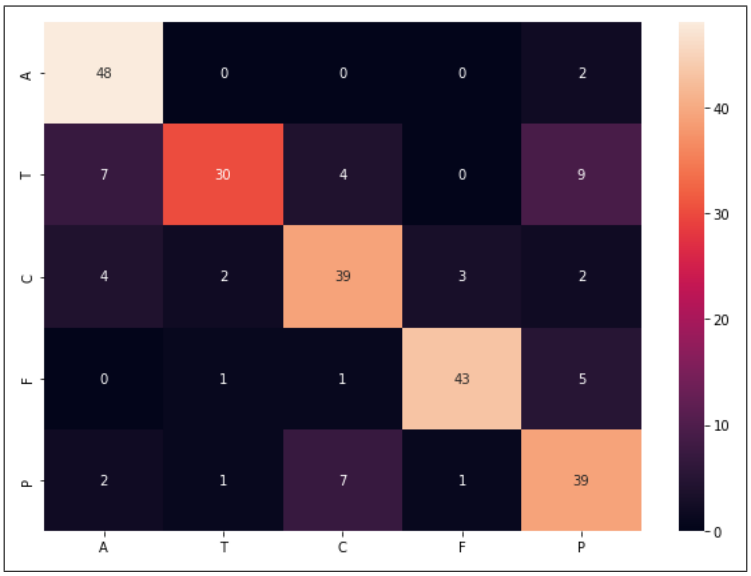

FIGURE 8. Confusion matrix of imbalanced dataset with customised VGG16 testing

on the features of the available dataset, Penicillium is the class with the greatest training data, thus we can infer this as a factor influencing the classification results. The training data is down-sampled throughout the subsequent training. This strategy was used to eliminate data bias towards specific classes produced by an imbalance in the amount of training data. The next experiment was using 100 training images for each class and 50 testing images for each class. The divisions of training and testing images data can be seen in Table 5.

TABLE 5. Balanced dataset

Genus	Training imagesl	Testing images	Total
Aspergillus	100	50	150
Cladosporium	100	50	150
Fusarium	100	50	150
Penicillium	100	50	150
Trichoderma	100	50	150

The confusion matrix in Figure 9 shows that the result model was successful in minimizing bias. To determine which model works better, the performance of the imbalanced dataset with the baseline architecture is compared to the balanced dataset with customised VGG16 . The performance of each model is shown in Table 6 and Table 7. In Table 6, it can be seen the performance obtained in the first experiment. The highest precision is obtained by the Fusarium class, which is 0.91, meaning that Fusarium gets the highest level of accuracy in making identification, while Penicillium gets a Precision value of 0.58 which is the lowest value compared to other classes, which means that Penicillium gets the lowest level of accuracy in making identification correctly. Aspergillus got the highest recall value, which was 0.96, meaning that the success rate of the model in re-finding the Aspergillus class was quite high compared to other classes, such as the Penicillium class, which only got a recall value of 0.68. The accuracy obtained by the first experiment was 79.6. In Table 7, it can be seen that the Cladosporium class obtained the highest precision with a value of 0.80, while the class with the lowest precision was obtained by

Penicillium, which was 0.60. The Fusarium classes obtained the highest recall value, with a value of 0.9, while the Penicillium class obtained the lowest recall, with a value of 0.50. The accuracy obtained by second experiment was 66.8

TABLE 6. The result of customized VGG16 with imbalanced dataset

Class	Precision	Recall	F1 Score
Aspergillus(A)	0.78	0.96	0.86
Cladosporium(C)	0.76	0.78	0.77
Fusarium(F)	0.91	0.86	0.72
Penicillium(P)	0.58	0.68	0.88
Trichoderma(T)	0.88	0.88	0.88

TABLE 7. The result of customized VGG16 with balanced dataset

Class	Precision	Recall	F1 Score
Aspergillus(A)	0.71	0.78	0.74
Cladosporium(C)	0.80	0.42	0.55
Fusarium(F)	0.63	0.90	0.74
Penicillium(P)	0.60	0.50	0.54
Trichoderma(T)	0.64	0.78	0.69

Those experiments showed that the model's performance with imbalanced data training is better than the down-sampling method. It can be seen by the values of recall, precision, and F1-score obtained, which were not higher than the first model, which was made with imbalanced data training. This result indicates that although down-sampling was successful in eliminating bias, however the down-sampling method does not improve performance but can cause a lack of data[26] and omit important information[27].

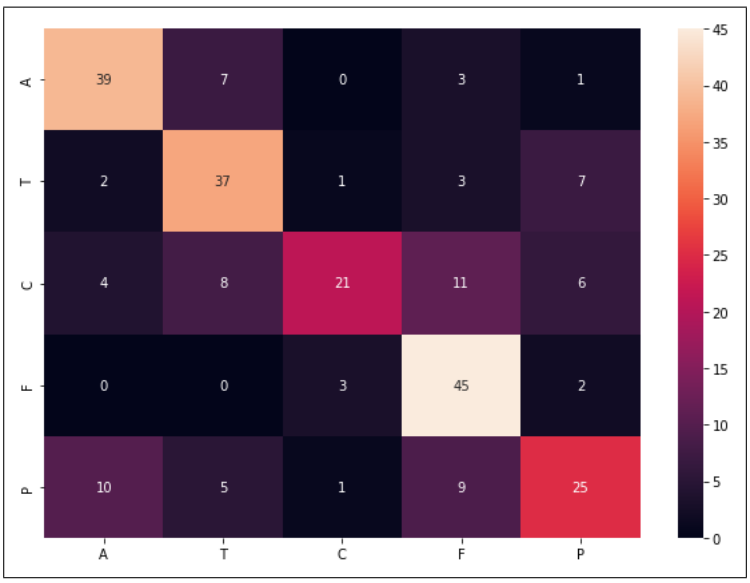

FIGURE 9. Confusion matrix of balanced dataset with customised VGG16 testing

Then an analysis of the accuracy and loss curves for each model is shown in Figure 10 and Figure 11. Both balanced and imbalanced models produce curves that tend to be similar. There is a large gap between performance during training and validation.

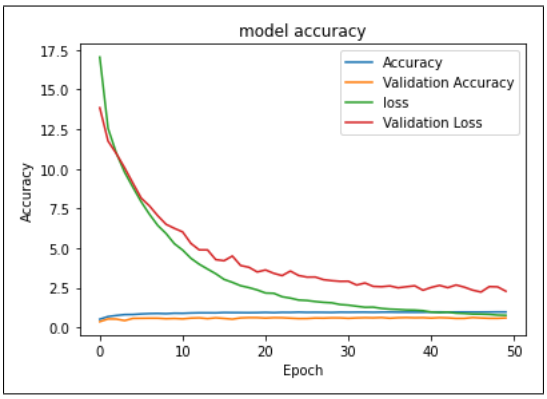

FIGURE 10. Accuracy and loss curve of imbalanced training

The training process was only carried out until epoch 50 because there was no significant increase in the curve after the 50th epoch. The learning process during the training did not go well. We can observe from those experiments that the models are overfitted. Overfitting happen when a model performs well with training data but poorly with test data[28]. Overfitting is mostly caused by noise learning on the training dataset or the training dataset is too small[29].

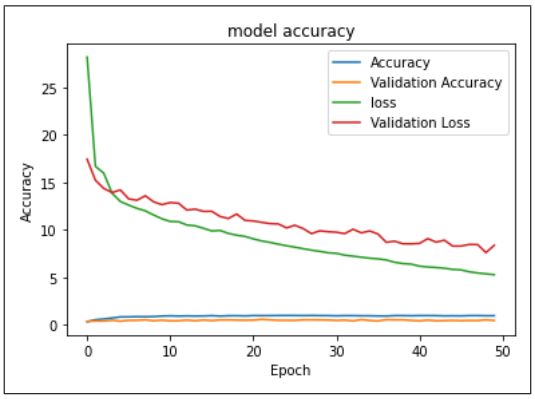

FIGURE 11. Accuracy and loss curve of balanced training

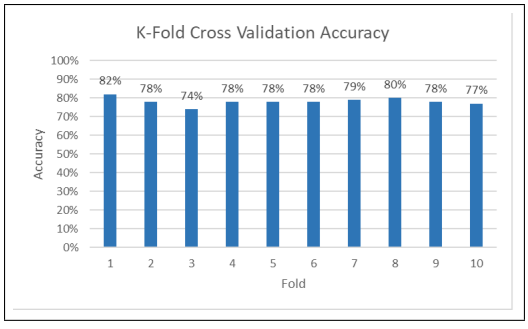

FIGURE 12. K-Fold Cross Validation accuracy

K-Fold cross validation is a common method to prevent overfitting of the model[30]. In the next experiment, we used k-fold cross validation to generate models from the imbalanced dataset with customised VGG16. We used 10-Fold cross validation, so we have 10 models as a result. As we can see in Figure 12 the model with the best performance

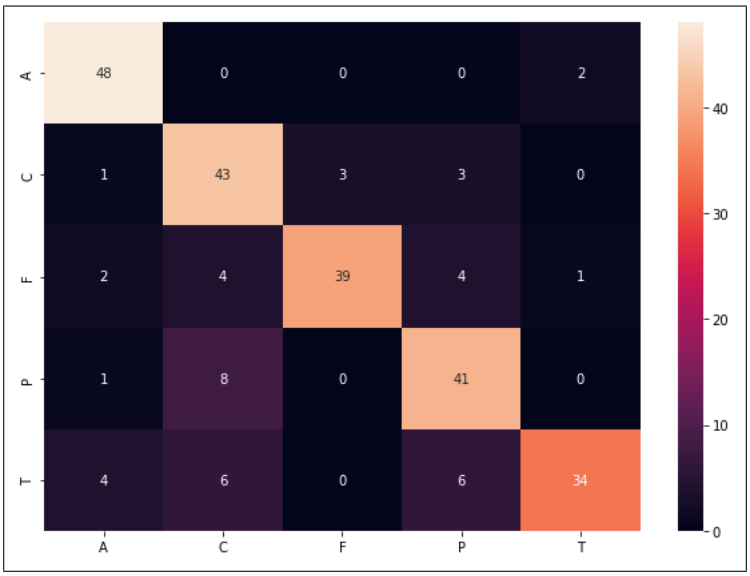

FIGURE 13. Confusion matrix of K-Fold Cross Validation method testing

was obtained by the 1th model. The 1th model obtain 82% of accuracy. There is slightly increase from the imbalanced model accuracy. Confusion matrix and performance of testing with the 1th Fold model can be seen in Figure 13.

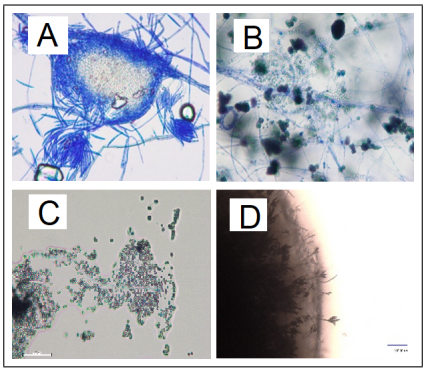

FIGURE 14. The example of images with noises(A), blurred image(B), incomplete fungi structure(C), bad lighting(D)

Although not significant, there is an increase in performance by applying k-fold cross validation to training processes. K-fold cross validation succeeded in producing the best combination for the distribution of training data and testing data. There was enhancement of accuracy obtained by third experiment. The accuracy for the third experiment was 82%. Difficulties faced include available dataset. Noise occurred in many of the images together with noises, blurred image, incomplete fungi structures captured, and bad lighting. examples of images with can be seen in Figure 14. As is known, that the quality of the image dataset will greatly affect the performance of the model. Thus, image enhancing techniques including histograms, fuzzy logic, and optimization methods can be used in future studies to address the issue of low-quality image datasets[31].

4. **Conclusion.** In applying computer vision for fungus identification, Deep Learning is a method that is considered more appropriate and efficient than using the classic machine learning method. Datasets in microscopic images have their difficulties compared

to non-microscopic objects. This study established a model for identifying fungi of the Aspergillus, Trichoderma, Cladosporium, Fusarium, and Penicillium genera. Four models were produced: an imbalanced data with baseline architecture, an imbalanced data with customised VGG16 model, a balanced data with customised VGG16 model, and an imbalanced data with customised VGG16 model with k-fold cross validation. The result obtained is models with customised VGG16 perform better than baseline architecture, and imbalanced dataset model perform better than balanced dataset model. It turns out that the down-sampling method has not improved model performance or eliminated bias in making decisions during classification. Besides that, the k-fold cross validation method has also been carried out to improve the performance of the resulting overfitting model. This method slightly improves the performance of the model.

REFERENCES

[1] J. S. Pan, H. C. Huang, L. C. Jain, and W. C. Fang (eds.), *Intelligent Multimedia Data Hiding: New Directions*, Springer, Berlin-Heidelberg, Germany, 2007.

[2] T. T. Bladt, J. C. Frisvad, P. B. Knudsen, and T. O. Larsen, Anticancer and Antifungal Compounds from Aspergillus, Penicillium and Other Filamentous Fungi, *Molecules*, vol.18, no.9, 2013.

[3] C. Li, J. Zhou, G. Du, J. Chen, S. Takahashi, and S. Liu, Developing Aspergillus Niger as a Cell Factory for Food Enzyme Production, *Biotechnol. Adv.*, vol.44, 2020.

[4] R. Tyśkiewicz, A. Nowak, E. Ozimek, and J. Jaroszuk-ściseł, Trichoderma: The Current Status of Its Application in Agriculture for the Biocontrol of Fungal Phytopathogens and Stimulation of Plant Growth, *Int. J. Mol. Sci.*, vol.23, no. 4, 2022.

[5] J. Poveda, Trichoderma as Biocontrol Agent Against Pests: New Uses for a Mycoparasite, *Biol. Control*, vol.159, 2021.

[6] R. P. A. Pramesti, et. al, Morphological And Textural Feature Extractions From Fungi Images For Development Of Automated Morphology-Based Fungi Identification System, *J. Bioteknol. BIOSAINS Indones.*, vol.9, no. 2, 2022.

[7] S. Reema Sree, S. B. Vyshnavi, and N. Jayapandian, Real-World Application of Machine Learning and Deep Learning, *Proc. 2nd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2019*,pp. 1069–1073, 2019.

[8] Y. Li, Research and Application of Deep Learning in Image Recognition, *2022 IEEE 2nd Int. Conf. Power, Electron. Comput. Appl. ICPECA 2022*, pp. 994–999, 2022.

[9] N. Ahmed, A. Yigit, Z. Isik, and A. Alpkocak, Identification of Leukemia Subtypes from Microscopic Images using Convolutional Neural Network, *Diagnostics*, vol. 9, no. 3, 2019.

[10] P. H. C. Chen, et. al, An Augmented Reality Microscope with Real-time Artificial Intelligence Integration for Cancer Diagnosis, *Nat. Med.*, vol. 25, no. 9, pp. 1453–1457, 2019.

[11] K. Simonyan, and A. Zisserman, Very Deep Convolutional Networks for Large-scale Image Recognition, *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp.1–14, 2015.

[12] A. Lumini, and L. Nanni, Deep Learning and Transfer Learning Features for Plankton Classification, *Ecol. Inform.*, vol. 51, no. February, pp. 33–43, 2019.

[13] S. Alagu, and K. Bhoopathy Bagan, Classification Of Leukocytes Using Convolutional Neural Network And Transfer Learning In Microscopic Blood, *International Journal of Research and Analytical Reviews*, vol. 6, no. 2, 2019.

[14] Z. M. Chan, C. Y. Lau, and K. F. Thang, Visual Speech Recognition of Lips Images Using Convolutional Neural Network in VGG-M Model, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 11, no. 3, 2020.

[15] M. Elgendi, et. al, The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective, *Front. Med.*, vol. 8, no. March, pp. 1–12, 2021.

[16] C. Shorten , and T. M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning, *J. Big Data*, vol. 6, no. 1, 2019.

[17] N. E. Khalifa, M. Loey, and S. Mirjalili, A Comprehensive Survey of Recent Trends in Deep Learning for Digital Images Augmentation, *Artif. Intell. Rev.*, vol. 55, no. 3, pp. 2351–2377, 2022.

[18] R. Shashidhar, and S. Patilkulkarni, Visual Speech Recognition for Small Scale Dataset using VGG16 Convolution Neural Network, *Multimed. Tools Appl.*, vol. 80, no. 19, pp. 28941–28952, 2021.

[19] D. Albashish, R. Al-Sayyed, A. Abdullah, M. H. Ryalat, and N. Ahmad Almansour, Deep CNN Model based on VGG16 for Breast Cancer Classification, *2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc.*, pp. 805–810, 2021.

[20] A. Krishnaswamy Rangarajan, and R. Purushothaman, Disease Classification in Eggplant Using Pre-trained VGG16 and MSVM, *Scientific Reports*, vol. 10, no. 1, pp. 1–11, 2020.

[21] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, Identification of Plant-leaf Diseases using CNN and Transfer-learning Approach, *Electronics*, vol. 10, no.20, 2021.

[22] J. M. Valverde , et. al, Transfer Learning in Magnetic Resonance Brain Imaging: A Systematic Review, *J. Imaging*, vol. 7, no. 4, pp. 1–21, 2021.

[23] M. Humayun, R. Sujatha, S. N. Almuayqil, and N. Z. Jhanjhi, A Transfer Learning Approach with a Convolutional Neural Network for the Classification of Lung Carcinoma, *Healthc*, vol. 10, no.6, oo. 1–15, 2022.

[24] K. Punitha, and S. Chitra, Software Defect Prediction using Software Metrics - A survey, *2013 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2013*, pp.555–558, 2013.

[25] J. A. Cottam, N. C. Heller, C. L. Ebsch, R. Deshmukh, P. MacKey, and G. Chin, Evaluation of Alignment: Precision, Recall, Weighting and Limitations, *Proc. - 2020 IEEE Int. Conf. Big Data, Big Data 2020*, pp.2513–2519, 2020.

[26] S. Suh, H. Lee, J. Jo, P. Lukowicz, and Y. O. Lee, Generative Oversampling Method for Imbalanced Data on Bearing Fault Detection and Diagnosis, *Applied Sciences*,vol. 9, no. 4, 2019.

[27] R. Rustogi , and A. Prasad, Swift Imbalance Data Classification using SMOTE and Extreme Learning Machine, *ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc.*, pp.1–6, 2019.

[28] S. Salman , and X. Liu, Overfitting Mechanism and Avoidance in Deep Neural Networks, *ICCIDS 2019 - 2nd Int. Conf. Comput. Intell. Data Sci. Proc.*, 2019, [Online]. Available: http://arxiv.org/abs/1901.06566.

[29] X. Ying, An Overview of Overfitting and its Solutions *Journal of Physics: Conference Series PAPER*, vol. 1168, no. 2, 2019.

[30] P. Charilaou , and R. Battat, Machine Learning Models and Over-fitting Considerations, *World J. Gastroenterol.*, vol. 28, no. 5, pp. 605–607, 2022.

[31] T. N. Tu, A new operator of enhancing dark images and application in binarizing dark images, *Journal of Information Hiding and Multimedia Signal Processing.*, vol. 15, no. 1, 2024.