IT Independent Journal
Vol. 13 No. 4, April (2025), pp. 379-388

ISSN 2301-8984 (Print), 2809-1884 (Online)
Published by Institute of Computer Science

Classification of mushroom types based on digital image processing using convolutional neural network

Ira Puspita Sari[1], Luluk Elvitaria[2]
[1,2]Informatics Engineering, Faculty of Engineering, Universitas Abdurrab, Pekanbaru, Indonesia

ARTICLE INFO

Article history:

Received Mar 24, 2025
Revised Mar 28, 2025
Accepted Apr 7, 2025

Keywords:

Classification;
CNN;
EfficientNet-B4;
Mushroom.

ABSTRACT

In this research, a classification of mushroom types based on digital image processing using a Convolutional Neural Network (CNN) is conducted. The method employs the EfficientNet-B4 architecture as the base model utilizing transfer learning and fine-tuning processes. The dataset consists of 3000 types of mushrooms, each categorized into 10 classes with 300 images per class. The CNN model is implemented using the Python programming language on Google Colab editor. Performance evaluation is carried out using accuracy, precision, recall, and F1-Score metrics to measure the model's performance. A comparison is made between all models with various training parameters, including identical and different settings. Additionally, the ratio of data splits, whether identical or different, is considered. Model 1, which utilizes a custom freeze layer and a data split ratio of 80% for training, 10% validation, and 10% testing, achieved the highest accuracy (90.00%), precision (90.09%), recall (89.63%), and F1-Score (89.59%) compared to other models. Therefore the implementation of a custom freeze layer to reduce the number of trainable parameters significantly impacts the accuracy level of the trained and tested model. Moreover, the determination of the data split ratio also slightly influences the accuracy level of the trained and tested model.

Corresponding Author:

Ira Puspita Sari,
Department of Informatics Engineering,
Universitas Abdurrab,
Jl. Riau Ujung No. 73 Pekanbaru, Riau, 28291, Indonesia.
Email: ira.puspita.sari@univrab.ac.id

1. INTRODUCTION

Fungi known by the Latin, are plants that do not have chlorophyll (Khosi'in, 2021). There are many mushrooms spread around the world, in addition to some being dangerous or poisonous, there are also several types of mushrooms that can be consumed by humans. Because of the diversity of fungi, and the morphology of some types or species of fungi that are dangerous or poisonous, some resemble mushrooms that can be consumed (Huang et al., 2023). To be able to better introduce to the public several species of fungi, to be classified between non-toxic mushroom species and poisonous mushrooms, a system is needed that can classify the type or species of the fungus. Deep learning is one of the things that is often discussed in the world of machine learning (Ghoshal et al., 2023). This deep learning method has significant or complex results in image recognition, namely Convolutional Neural Network (CNN). CNN can learn important patterns contained in input data, so it can recognize objects well enough so that image/image learning functions are easier to implement (Lestandy et al., 2022).

Based on the description above, a good solution is needed to help classify types or species in fungi easily and efficiently, namely by using the system "Classification of Fungi based on Digital Image Processing using Convolutional Neural Network (CNN)". The system is expected to determine the accuracy of the resulting method, how well CNN can classify mushroom images by types or species and can introduce the public to several species of non-toxic mushrooms and poisonous mushrooms (Mbakop et al., 2023). The dataset used in this study includes 10 mushroom species consisting of poisonous and non-toxic types, with a total of 3000 images. Each mushroom class has 300 images divided into three parts for training, validation, and testing, with several different data sharing scenarios (e.g. 80%-10%-10% or 70%-20%-10%). The diversity of this dataset is sufficient to distinguish between toxic and non-toxic mushrooms within the scope of the selected species. However, to reflect the real conditions in the field more broadly, a larger increase in the number and variety of species is needed, given that there are more than 100,000 species of fungi that have been identified, and about 1.5 million species are estimated to exist in the world. The purpose of the study is to be able to find out how well the CNN method uses the EfficientNet-B4 architecture in classifying species or species images in fungi, find out the results of the accuracy level of image classification of species types in fungi, and introduce digital image processing technology and deep learning algorithms that can be used in the field of mycology, to make it easier for mycologists to work and research the diversity of fungi that can be classified with digital image processing. To improve the reliability of the model, the dataset should be expanded with variations in environmental factors through data augmentation such as adjusting lighting (brightness, contrast), changing the shooting angle (rotation, flipping), and adding noise or blur to mimic real-world conditions (Ve$nugopal e$t al., 2023).

2. RESEARCH METHOD

The main problem that becomes the background in identifying or formulating problems and becoming the object of research is the difficulty in identifying types (species) in fungi automatically or efficiently (Ragab & Albukhari, 2022). Traditional methods used to classify types (species) in fungi through visual observation and morphological description tend to be time-consuming and less effective in distinguishing types (species) in similar fungi, to make it easier for mycologists to work and examine the diversity of fungi to classify, it can use digital image processing technology and deep learning algorithms that will be used for image classification in species or types of such mushrooms (Chen et al., 2023). This research applies a data augmentation strategy by resizing the image to 380x380 pixels, converting the color scheme to RGB, and dividing the dataset into various ratios for training, validation, and testing to evaluate model performance. The model used is EfficientNet-B4 with transfer learning and fine-tuning methods to improve the accuracy of mold classification. EfficientNet-B4 was chosen as the base model due to its efficiency in parameter usage and computation as well as its high accuracy, as supported by the literature and preliminary tests. This model uses the strategy of “compound scaling,” which allows for optimal scaling, resulting in an accuracy of up to 90% in this study. Compared to other models such as GPipe, EfficientNet-B4 has smaller parameters and higher speed. The use of transfer learning and fine-tuning methods also speeds up training and improves classification performance. Experimental results show that the configuration with freeze layer (custom) provides better generalization than non-custom models, making it a good choice for image classification tasks.

The dataset is grouped based on imagery of species in fungi. There are ten categories for each species in the fungus, namely agaricus bisporus, amanita phalloides, boletus edulis, entoloma sinuatum, suillus luteus, cortinarius rubellus, hygrocybe punicea, lactarius glyciosmus, russula brevipes, and auricularia auricula-judae (Dai et al., 2023). For public image dataset acquisition is a technique used to collect image datasets from public sources available online. This technique utilizes images data sources already available on the internet such as websites, forums, social networks, and open data sources. In this technique image dataset retrieval is done by downloading or copying images from existing data sources (Abraham et al., 2023). The dataset is obtained from images or images of fungi based on their species obtained from one of the sites in Kaggle (https://www.kaggle.com/code/yelloworld/mushrooms-classification/input). Because the dataset of species imagery on fungi obtained is still lacking and adapts to classes in predetermined species or types of fungi, the researchers also obtained images of mushrooms by species from one of the sites, a site about audience science projects and an online social network for nature lovers, audience scientists, and biologists built on the concept of mapping and sharing observations of biodiversity from around the world, One of them is this site contains various types of images on fungi taken by scientists or mycologists (https://www.inaturalist.org/). The types of images/images used in this research dataset are RGB image types (Red, Green, Blue) (López et al., 2022). RGB is a digital image consisting of three basic colors, namely red, green, and blue. Implementation in mushroom image classification research refers to the process of applying or applying the results of research that has been carried out in the form of a mushroom classification model based on digital image processing using a Convolutional Neural Network (CNN) into the system (Yamamoto et al., 2022).

System Design

At this stage, researchers will first design or design a model on the system in this study using a flowchart, which contains the stages of how the system works and facilitates the stages of making or coding models on the system (Rao et al., 2023) (Pratiwi et al., 2021). The stages in the system design flow diagram in the study above can be simplified into several stages or processes. The initial stage carried out by preparing an image dataset is to select and prepare an image dataset with 10 classes of types or species in fungi (some are non-toxic and some are toxic) that will later be used for training and testing. Image datasets can be retrieved from dataset provider sites for types or species in fungi. Then, the data preparation process and CNN model were carried out (Justaniah et al., 2022) (Miller & Buyck, 2002).

The next process is pre-processing, which is the random division (split) of data on the dataset which will be divided into three parts of data, namely training data (80%), validation data (10%), and testing data (10%) and also split data with training data (70%), validation data (20%), and testing data (10%) to later compare the accuracy of several models trained and tested with different data split ratios. The next process is modeling with the EfficienNet-B4 architecture with transfer learning and fine-tuning techniques. After determining the weight of the random data, enter the CNN training stage with the EfficientNet-B4 architecture, which is to carry out the learning or training process on the CNN model using the EfficientNet-B4 architecture as the basis. The CNN training process with the EfficientNet-B4 architecture has previously carried out transfer learning and fine-tuning techniques to improve model performance (Zhao et al., 2022).

In the process of such training if the number of epochs is exhausted the training will be stopped and the model will be saved. However, if the number of epochs has not been exhausted, the epochs will continue to run and the number of epochs will decrease one by one which has previously been determined how many epochs (epochs are the size of training as much as 1 cycle) (Acharya et al., 2022) (Apriayani, 2023). The system will check again whether there are still epochs left if there are then the training process is carried out again (with the EfficientNet-B4 architecture), while if the number of epochs has been exhausted (there are no epochs left) then the model has been trained (Zhu et al., 2021). The trained model will be tested using images that have been prepared on split data, namely previous test data. Then the system will perform calculations or classifications using the model that has been trained earlier. This system will end by issuing results in the form of sum values from accuracy, precision, recall, and F1-Score (Purkayastha & Mohapatra, 2023) (Haksoro & Setiawan, 2021).

Coding

The system design on CNN uses a predefined architecture (EfficientNet-B4) with the chosen programming language, namely Python language, and uses the Tensor Flow library (Denandra et al., 2023). The use of the Google Colab platform is as a means of designing, namely modeling or coding the system and using Google Drive as a storage medium. After training the CNN model, researchers will interpret the results or convey the results of research obtained from the model, the results of which are interpreted by the problem formulation and objectives of the research that has been set. This is very useful so that the intent of the model used can be conveyed well in this study (Rahman & Das, 2022) (Kaplale et al., 2022). After the model is trained and gets the final result, the results are evaluated on the model, namely from the performance of the CNN model with the EfficientNet-B4 architecture that has been predetermined whether the model can perform good or optimal learning performance in the training and testing (Mosayyebi et al., 2023) (Baykal et al., 2020).

3. RESULTS AND DISCUSSIONS

Researchers used the EfficientNet-B4 architecture as the basis for the mushroom-type image classification model. EfficientNet-B4 is one of the efficient CNN architectures that are quite good at overcoming image classification problems. Researchers utilize transfer learning and fine-tuning methods on the EfficientNet-B4 architecture. Figure 1 is EfficientNet-B4 Implementation Program Code.

```python
import keras
from keras.layers import Dropout
from keras.models import Sequential
from keras.layers import Dense, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.applications.efficientnet import EfficientNetB4


base_model = EfficientNetB4 (include_top= False,
                       weights = 'imagenet',
                       input_shape = (380,380,3))

#lihat struktur base_model
base_model.summary()
```

Figure 1. E$fficie$ntNe$t-B4 Imple$me$ntation Program Code$

Using the code, the researchers imported EfficientNet-B4 as a base model and printed a summary of the model. Next, the researcher can proceed to add additional layers to the base model as needed for specific image classification tasks. Researchers freeze the basic model layer because the model is compared for the number of parameters between freeze (custom) and without freeze and freeze (non-custom) and freeze. By setting layer trainable = False, layers before the layer named 'block7a_expand_conv' will be frozen while layers after it can still be trained. This is useful if you want to set a certain part of the model as a pretrained part or want to maintain the weight already in that layer or other words by applying freeze to the base model layer, the number of parameters used during training will be reduced. Table 1 describe the comparison bumber of freeze with custom and no freeze parameters.

Table 1. Comparison the number of freeze (custom) and no freeze parameteres

Model	Data Split Ratio	Parameter	Freeze (custom)	Without Freeze
1	Train (80%)	Total	83,777,193	
	Validation (10%)	Trainable	71,379,262	83,651,474
	Test (10%)	Non-Trainable	12,397,931	125,719
2	Train (70%)	Total	83,777,193	
	Validation (20%)	Trainable	71,379,262	83,651,474
	Test (10%)	Non-Trainable	12,397,931	125,719

Next freeze (non-custom), which is to set so that all layers in the base_model model cannot be trained or updated during the training process. A for loop that iterates through each layer in the base_model. In this loop, each layer is set to non-trainable (trainable = False) by setting the trainable property of each layer to False. This means that when the model_new model is built and trained, the layers in the base_model will not be updated based on the gradient obtained during training. In other words, that is, the extraction feature is frozen (following the default architecture). Once all layers in the base_model are set to untrainable, we print a summary of the model new. Table 2 describe the comparison bumber of freeze with no custom and no freeze parameters.

Table 2. Comparison the number of freeze (non-custom) and no freeze parameters

Model	Data Split Ratio	Parameter	Freeze (non-custom)	Without Freeze
3	Train (80%)	Total	83,777,193	
	Validation (10%)	Trainable	66,102,858	83,651,474
	Test (10%)	Non-Trainable	17,674,335	125,719
4	Train (70%)	Total	83,777,193	
	Validation (20%)	Trainable	66,102,858	83,651,474
	Test (10%)	Non-Trainable	17,674,335	125,719

The results of the training process that has been carried out, can be analyzed comparison between model 1 with different models 2, 3, and 4. So, it can be seen in the following table 3:

Table 3. The valuable of training and testing on model 1

Training						
Epoch to-	Loss	Accuracy	Validation Loss	Validation Accuracy	Data Split Ratio	Number of Images
1	1.7388	0.4650	1.0498	0.6667	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
2	0.5762	0.8658	0.5717	0.8233	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
3	0.2210	0.9642	0.3812	0.8667	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
4	0.0982	0.9904	0.3205	0.8933	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
5	0.0495	0.9975	0.3091	0.9000	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
6	0.0377	0.9946	0.2954	0.9067	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Te$st (300)
7	0.0264	0.9979	0.3300	0.9067	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Tst (300)
8	0.0180	0.9992	0.3103	0.9000	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
9	0.0123	0.9992	0.3077	0.9100	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)
10	0.0113	0.9992	0.2906	0.9133	Train (80%) Validation (10%) Test (10%)	Train (2400) Validation (300) Test (300)

Tsting			
Metrics	Value	Data Split Ratio	Number of Images
Accuracy	0.9 (90.000%)	Test (10%)	Test (300)
Average *Precision*	0.9009 (90.09%)	Test (10%)	Test (300)
Average *Recall*	0.8963 (89.63%)	Test (10%)	Test (300)
Average *F1-Score*	0.8959 (89.59%)	Test (10%)	Test (300)

The results of training and testing can be concluded from the table data above that the researchers provide, namely model 1, with the ratio of split data for the training (80%), validation (10%), and test (10%). This model before training, freezes the layer (custom) on the model new. Training is carried out in 10 epochs. The loss and accuracy of the model are calculated at each epoch. At the beginning of training, the loss was 1.7388 and the accuracy was 0.4650. As the epoch progresses, the loss decreases quite significantly, and the accuracy increases. At the last epoch, the loss was 0.0113 and the accuracy reached 0.9992.

Classification Test

For the classification test using test data as many as 30 mushroom images with each class consisting of 3 test data images which are obtained on sites containing images of various types of fungi (https://www.inaturalist.org/). This classification test data consists of various images that the model has not previously seen during the training and validation process. This test data aims to evaluate the extent to which the model can perform classification or grouping with good accuracy on new or previously unknown data. The classification testing phase starts by presenting or displaying one by one the type image on the mushroom into a trained model (which uses a model with high accuracy, namely model 1). Then, the model will make predictions on each image and classify them or classify them as fungi of a certain type or class. For the prediction results the model will be compared with the class name (original) and type of toxic or not (original) of the mushroom image to determine the degree of congruence between the prediction and reality or the original. The results of the classification test can be seen in the following table 4:

Table 4. Classification test result

Image	Original Class	Prediction Class	Poisonous/ Non (Original)	Poisonous/ Non (Prediction)
	Agaricus Bisporus	Agaricus Bisporus	Non-poisonous	Non-poisonous
	Agaricus Bisporus	Agaricus Bisporus	Non-poisonous	Non-poisonous
	Agaricus Bisporus	Agaricus Bisporus	Non-poisonous	Non-poisonous
	Amanita Phalloides	Amanita Phalloides	Poisonous	Poisonous
	Amanita Phalloides	Amanita Phalloides	Poisonous	Poisonous
	Amanita Phalloides	Entoloma Sinuatum	Poisonous	Poisonous
	Auricularia Auricula-Judae	Auricularia Auricula-Judae	Non-poisonous	Non-poisonous
	Auricularia Auricula-Judae	Auricularia Auricula-Judae	Non-poisonous	Non-poisonous
	Auricularia Auricula-Judae	Auricularia Auricula-Judae	Non-poisonous	Non-poisonous
	Boletus Edulis	Boletus Edulis	Non-poisonous	Non-poisonous
	Boletus Edulis	Boletus Edulis	Non-poisonous	Non-poisonous

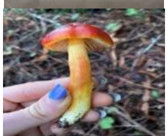
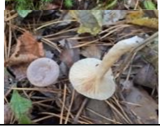
Image	Original Class	Prediction Class	Poisonous/ Non (Original)	Poisonous/ Non (Prediction)
	Boletus Edulis	Boletus Edulis	Non-poisonous	Non-poisonous
	Cortinarius Rubellus	Cortinarius Rubellus	Poisonous	Poisonous
	Cortinarius Rubellus	Entoloma Sinuatum	Poisonous	Poisonous
	Cortinarius Rubellus	Cortinarius Rubellus	Poisonous	Poisonous
	Entoloma Sinuatum	Entoloma Sinuatum	Poisonous	Poisonous
	Entoloma Sinuatum	Entoloma Sinuatum	Poisonous	Poisonous
	Entoloma Sinuatum	Entoloma Sinuatum	Poisonous	Poisonous
	Hygrocybe Punicea	Hygrocybe Punicea	Poisonous	Poisonous
	Hygrocybe Punicea	Hygrocybe Punicea	Poisonous	Poisonous
	Hygrocybe Punicea	Hygrocybe Punicea	Poisonous	Poisonous
	Lactarius Glyciosmus	Lactarius Glyciosmus	Non-poisonous	Non-poisonous
	Lactarius Glyciosmus	Lactarius Glyciosmus	Non-poisonous	Non-poisonous
	Lactarius Glyciosmus	Lactarius Glyciosmus	Non-poisonous	Non-poisonous

Image	Original Class	Prediction Class	Poisonous/ Non (Original)	Poisonous/ Non (Prediction)
	Russula Brevipes	Russula Brevipes	Poisonous	Poisonous
	Russula Brevipes	Russula Brevipes	Poisonous	Poisonous
	Russula Brevipes	Russula Brevipes	Poisonous	Poisonous
	Suillus Luteus	Suillus Luteus	Non-poisonous	Non-poisonous
	Suillus Luteus	Suillus Luteus	Non-poisonous	Non-poisonous
	Suillus Luteus	Auricularia Auricula-Judae	Non-poisonous	Non-poisonous

Evaluation of Result

After getting the results of the training and prediction process, researchers then analyze important metrics, namely loss, accuracy, validation loss, and validation accuracy. To make the analysis easy to do, the researchers use the matplotlib library to produce a graph visualization, namely a comparison between loss and validation loss, as well as accuracy and validation accuracy. The results of the comparison of accuracy, precision, recall, and F1 score in models can be seen in the following table 5:

Table 5. Comparison of accuracy, precision, recall, and f1 score in models

Model	Data Split Ratio	*Freeze Layer*	*Accuracy*	*Precision*	*Recall*	*F1-Score*
1	(80,10,10)	*Custom*	90.000%	90.09%	89.63%	89.59%
2	(70,20,10)	*custom*	89.667%	90.87%	88.99%	89.55%
3	(80,10,10)	*non-custom*	82.667%	86.28%	82.09%	82.54%
4	(70,20,10)	*non-custom*	80.333%	83.42%	80.52%	80.03%

When viewed from the testing side, namely the comparison between training data and test data against all models with the same or different training parameters and also the data split ratio is the same and different, for model 1 has the highest level of accuracy (90,000%), precision (90.09%), recall (89.63%), and F1-Score (89.59%) compared to other models. This shows that using the same architecture for all models, namely EfficientNet-B4, freeze layer (custom), and with the ratio of split data train (80%), validation (10%), test (10%) has a better ability to generalize to the model compared to the application of freeze layer (non-custom), and with the ratio of split data train (70%), validation (20%), test (10%) to other models.

Based on the comparison of several models that have been trained and tested, it can be evaluated that the best model with high accuracy (training and testing) is a model with the application of freeze layer (custom) found in model 1 and model 2, so for this study that affects the level of accuracy of the model that has been trained and tested, namely based on the application of freeze layer (custom) to reduce the number of parameters to be trained and the determination of the split data ratio as well slightly affects the accuracy level of the model that has been trained and tested (can be seen in model 1 and model 2). In addition to evaluating training, evaluation of classification testing was also carried out on image data of new or unseen mushroom species. For the classification test, three reading errors occurred in the mushroom classification test. Misreadings are that the model provides inaccurate predictions for three of the 30 image data on the type of fungus. However, this error is normal in model testing, because each classification model has limitations in its generalizability.

4. CONCLUSION

In this study, the types of fungi consisting of 10 classes (categories) are classified based on image processing using Convolutional Neural Network (CNN) using the EfficientNet-B4 architecture which has previously been carried out transfer learning and fine-tuning on the architecture. The researchers assigned the dataset to 3,000 images and assigned 300 images to each class or type of fungus. According to the results of the study, it can be seen that the use of the EfficientNet-B4 architecture results in good model performance. By using this architecture, a comparison between training data and test data is obtained against all models with the same or different training parameters, and the data split ratio is the same and different, for model 1 has the highest level of accuracy (90,000%), precision (90.09%), recall (89.63%), and F1-score (89.59%) compared to other models.

This shows that by using the same architecture for all models, namely EfficientNet-B4, freeze layer (custom), and with the ratio of split data train (80%), validation (10%), test (10%) has a better ability to generalize to the model compared to the application of freeze layer (non-custom), and with the ratio of split data train (70%), validation (20%), test (10%) to other models. Based on this research, has a major effect on the level of accuracy of models that have been trained and tested, namely based on the application of a frozen layer (custom) to reduce the number of parameters to be trained and the determination of split data ratios also slightly affects the level of accuracy of models that have been trained and tested. This research has relevance to the field of mycology as the CNN-based classification method used can assist in the identification of fungal species, including toxic and non-toxic ones. This is important in the study of fungi biodiversity as well as applications in the food and pharmaceutical industries. This model can also be used by mushroom researchers and farmers on a larger scale with some adjustments, such as expanding the dataset to be more diverse and optimizing the CNN architecture to improve accuracy. In addition, implementation in the form of AI-based software or applications can help rapid identification for farmers in accurately determining the type of mushroom, reducing the risk of consumption of toxic mushrooms, and increasing production efficiency. The results of this study show that the generalization of the model to fungal species outside the dataset used still faces some challenges, especially because some species have similar shapes that can cause prediction errors. To improve the generalization ability of the model, several strategies can be applied, such as enlarging and enriching the dataset with a wider variety of images, performing optimal hyperparameter tuning, and utilizing transfer learning from models that have been trained on more diverse datasets. In addition, the application of data augmentation techniques can also help the model to recognize a greater variety of shapes, thereby improving accuracy on data that has never been seen before.

REFERENCES

Abraham, B. S., Bhanu, D. G., Thomas, A., & Anitha, J. (2023). Comparative Analysis of Glaucoma Diagnosis on Retinal Fundus Images Using AI Techniques. *2023 4th International Conference on Signal Processing and Communication (ICSPC)*, 386–390.

Acharya, D., Guda, R. K. S., & Raovenkatajammalamadaka, K. (2022). Enhanced efficientnet network for classifying laparoscopy videos using transfer learning technique. *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–9.

Apriayani, S. (2023). *Pengembangan buku saku keanekaragaman hayati tumbuhan obat di desa mekar bersatu sebagai bahan ajar Biologi Kelas X MA Darul Muhajirin*.

Baykal, E., Dogan, H., Ercin, M. E., Ersoz, S., & Ekinci, M. (2020). Transfer learning with pre-trained deep convolutional neural networks for serous cell classification. *Multimedia Tools and Applications*, *79*, 15593–15611.

Chen, X., Pu, X., Chen, Z., Li, L., Zhao, K., Liu, H., & Zhu, H. (2023). Application of EfficientNet-B0 and GRU-based deep learning on classifying the colposcopy diagnosis of precancerous cervical lesions. *Cancer Medicine*, *12*(7), 8690–8699.

Dai, Q., Guo, Y., Li, Z., Song, S., Lyu, S., Sun, D., Wang, Y., & Chen, Z. (2023). Citrus Disease Image Generation and Classification Based on Improved FastGAN and EfficientNet-B5. *Agronomy*, *13*(4), 988.

Denandra, R., Fariza, A., & Prayogi, Y. R. (2023). Eye Disease Classification Based on Fundus Images Using Convolutional Neural Network. *2023 International Electronics Symposium (IES)*, 563–568.

Ghoshal, N., Anas, M., & Roy, S. S. (2023). Chest X-Ray Image Classification of Pneumonia Disease Using EfficientNet and InceptionV3. In *Deep Learning Applications in Image Analysis* (pp. 173–186). Springer.

Haksoro, E. I., & Setiawan, A. (2021). Pengenalan jamur yang dapat dikonsumsi menggunakan metode transfer learning pada convolutional neural network. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer*, *5*(2), 81–91.

Huang, Z., Su, L., Wu, J., & Chen, Y. (2023). Rock Image Classification Based on EfficientNet and Triplet Attention Mechanism. *Applied Sciences*, *13*(5), 3180.

Justaniah, E., Aldabbagh, G., Alhothali, A., & Abourokbah, N. (2022). Classifying Breast Density from Mammogram with Pretrained CNNs and Weighted Average Ensembles. *Applied Sciences*, *12*(11), 5599.

Kaplale, A. H., Rumondor, M., & Tangapo, A. M. (2022). Toksisitas Makrofungi Beracun dari Divisi Basidiomycota. *Journal of Biotechnology and Conservation in Wallacea*, *2*(1), 1–15.

Khosi'in, K. (2021). *BUKU AJAR MATA KULIAH KEANEKARAGAMAN MAKHLUK HIDUP (KINGDOM FUNGI) BERBASIS KEARIFAN LOKAL DI TAMAN NASIONAL BUKIT BARISAN SELATAN BENGKULU*. CV. ELSI PRO.

Lestandy, M., Hanifah, P. K., & Irfan, M. (2022). Deteksi Dini Kanker Payudara menggunakan Metode Convolution Neural Network (CNN). *Inspiration: Jurnal Teknologi Informasi Dan Komunikasi*, *12*(1), 65. https://doi.org/10.35585/inspir.v12i1.2667

López, A. R., Barea-Sepúlveda, M., Barbero, G. F., Ferreiro-González, M., López-Castillo, J. G., Palma, M., & Espada-Bellido, E. (2022). Essential mineral content (Fe, Mg, P, Mn, K, Ca, and Na) in five wild edible species of Lactarius mushrooms from Southern Spain and Northern Morocco: Reference to daily intake. *Journal of Fungi*, *8*(12), 1292.

Mbakop, S., Tagne, G., Lagache, A., Youcef-Toumi, K., & Merzouki, R. (2023). Integrated design of a bio-inspired soft gripper for mushrooms harvesting. *2023 IEEE International Conference on Soft Robotics (RoboSoft)*, 1–6.

Miller, S. L., & Buyck, B. (2002). Molecular phylogeny of the genus Russula in Europe with a comparison of modern infrageneric classifications. *Mycological Research*, *106*(3), 259–276.

Mosayyebi, F., Seyedarabi, H., & Afrouzian, R. (2023). Gender Recognition in Masked Facial Images Using Efficientnet and Transfer Learning Approach. *Available at SSRN 4328048*.

Pratiwi, H. A., Cahyanti, M., & Lamsani, M. (2021). Implementasi deep learning flower scanner menggunakan metode convolutional neural network. *Sebatik*, *25*(1), 124–130.

Purkayastha, R., & Mohapatra, S. (2023). A Conceptual Model for Analysis of Plant Diseases Through EfficientNet: Towards Precision Farming. In *Innovations in Machine and Deep Learning: Case Studies and Applications* (pp. 401–417). Springer.

Ragab, M., & Albukhari, A. (2022). Automated Artificial Intelligence Empowered Colorectal Cancer Detection and Classification Model. *Computers, Materials & Continua*, *72*(3).

Rahman, N., & Das, S. K. (2022). A Fusion of Three Custom-Tailored Deep Learning Architectures for Waste Classification. *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 1–6.

Rao, M. V. S. R., Puligundla, S., Ekkala, N. S., & Chebrolu, S. (2023). Image Classification of Ischemic Stroke Blood Clot Origin using Stacked EfficientNet-B0, VGG19 and ResNet-152. *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 304–310.

Venugopal, V., Raj, N. I., Nath, M. K., & Stephen, N. (2023). A deep neural network using modified EfficientNet for skin cancer detection in dermoscopic images. *Decision Analytics Journal*, 100278.

Yamamoto, S., Kinugasa, H., Hamada, K., Tomiya, M., Tanimoto, T., Ohto, A., Toda, A., Takei, D., Matsubara, M., & Suzuki, S. (2022). The diagnostic ability to classify neoplasias occurring in inflammatory bowel disease by artificial intelligence and endoscopists: A pilot study. *Journal of Gastroenterology and Hepatology*, *37*(8), 1610–1616.

Zhao, X., Li, K., Li, Y., Ma, J., & Zhang, L. (2022). Identification method of vegetable diseases based on transfer learning and attention mechanism. *Computers and Electronics in Agriculture*, *193*, 106703.

Zhu, C.-Y., Wang, Y.-K., Chen, H.-P., Gao, K.-L., Shu, C., Wang, J.-C., Yan, L.-F., Yang, Y.-G., Xie, F.-Y., & Liu, J. (2021). A deep learning based framework for diagnosing multiple skin diseases in a clinical environment. *Frontiers in Medicine*, *8*, 626369.