DETECTION OF HERBAL PLANT LEAVES USING A TRANSFER LEARNING APPROACH IN CNN WITH THE TENSORFLOW LIBRARY ON ANDROID

Iga Febyola Mambi [1] , Hanhan Maulana [2]

[1.2] Informatics Engineering Study Program , Indonesian Computer University
Jl. Dipati Ukur No.112-116, Lebakgede , District Coblong , Bandung City, West Java 40132
E-mail : iga.10118387@mahasiswa.unikom.ac.id [1] , hanhan@email.unikom.ac.id [2]

Abstract

There are approximately 30,000 types of herbal plants in Indonesia. However, only 1,200 types of herbal plants have been used by the Indonesian people as raw materials for alternative or traditional medicine. Herbal plants are not easy to recognize even though they often grow around the neighborhood. Because of the lack of public knowledge about herbal plants, herbal plants are not utilized. So this research will classify images on herbal plant leaves using the transfer learning method approach on CNN. In this research, the transfer learning method used is MobileNet V2 and InceptionV3 architecture. The image classification application applied is based on Android which is built with the programming language used in application development is python and JavaScript. The database used for data storage uses MySQL and utilizes the react native framework to build applications on the android platform. The results of the evaluation based on the confusion matrix of the MobileNet V2 architecture get an accuracy result as high as 96.63% and Inception V3 gets an accuracy of 97.2%, so that the performance results are superior to Inception V3 but the performance results of both are very good in classification.

Keywords : *Android , CNN Transfer Learning , Herbal Plants*

1. INTRODUCTION

Indonesian herbal plants are part of the wealth of traditional spices that play a significant role in medicine in Indonesia. The diversity of plants that thrive in this country is supported by the tropical climate and weather conditions that support their development [1] . There are approximately 30,000 types of herbal plants in Indonesia. However, only 1,200 types of herbal plants have been utilized by the Indonesian people as raw materials for alternative or traditional medicine .

Although they often grow around residential areas, herbal plants are not always easy to recognize. Lack of public knowledge about herbal plants results in their utilization being less than optimal. Among the various parts of herbal plants, leaves are one of the most frequently used as raw materials for traditional medicine [2] . However, many people are still unfamiliar with herbal leaves or medicinal leaves. This is due to the similarity in the shape of the leaves which makes them difficult to distinguish. In fact, if observed closely, each leaf has its own characteristics that distinguish it from other leaves. Therefore, special skills are needed to recognize herbal plants based on their leaves. Lack of public understanding of various types of herbal plants in detail is often an obstacle in utilizing these plants for traditional medicine, due to the difficulty in identifying them correctly [2] .

Digital image processing is a field that studies the formation, management and analysis of images in order to obtain information that can be used [3] . One implementation of image processing is detection. or classification [4] . Previous research conducted by Putra et al. focused on the classification of diseases and pests in rice plants using the transfer learning method. In the study, a comparison was made between several transfer learning models, namely MobileNet V2, NasNet Mobile, EfficientNet B7, Inception V3, VGG16, and a simple CNN-based model. The dataset used includes five types of diseases, three types of pests, and one category of healthy rice plants. Each data goes through a preprocessing and augmentation process before being used in model training. The results of the study show that the model with the Inception V3 architecture achieves very high accuracy, with an accuracy value of 97%, precision of 97%, sensitivity of 97%, specificity of 99%, and f1-score of 96%. However, this model has a very large number of parameters, which results in a heavier model weight. Based on a comparison between performance and the number of model parameters, MobileNet V2 provides quite good results with an accuracy of 96%, precision of 96%, sensitivity of 96%, specificity of 99%, and f1-score of 96%, making it a lighter alternative compared to Inception V3 [5] .

Therefore, this research was conducted by detecting images on herbal plant leaves using the *transfer learning method approach* on CNN so that the problem of manual identification processes that are often time-consuming, less accurate, and require special skills can be overcome with this research which seeks to improve the accuracy of herbal plant leaf image classification automatically. The *transfer learning method approach* is a method that uses a previously trained *network and uses it as a starting point for learning new tasks. Performing fine tuning network* with *transfer learning* is much faster and easier than training *the network* from scratch with randomly initialized weights [6] . In addition, the advantage of the transfer learning method lies in its ability to produce richer information for classification, because it has deeper layers compared to simple CNNs [7] [8] . In this study, the transfer learning method applied used the MobileNet V2 and Inception V3 architectures. MobileNet V2 was chosen because it has a fairly high level of accuracy and the main advantage is the number of training parameters that are smaller than other CNN models, so it requires lighter computing power [9] . Meanwhile, *Inception* V3 is a variant of the *deep learning architecture* built by Google. This architecture has more precise parameters because it is able to reduce the value of its parameters by distributing its weights into several *multi-layers.* [10] . Based on the advantages of each method, it is hoped that it can be known which *transfer learning method is the best.* in carrying out detection herbal plant leaves so that it can help in identifying herbal plant species quickly and accurately.

2. METHODOLOGY

2.1 Research methods

This research method uses a descriptive method. The descriptive method is a research method that aims to provide a systematic, factual, and accurate description and analysis of a particular phenomenon or problem. This study aims to describe the stages of Android-based software development using the *transfer learning approach* on CNN for detecting herbal plant leaves. The stages in this study include data collection and software development.

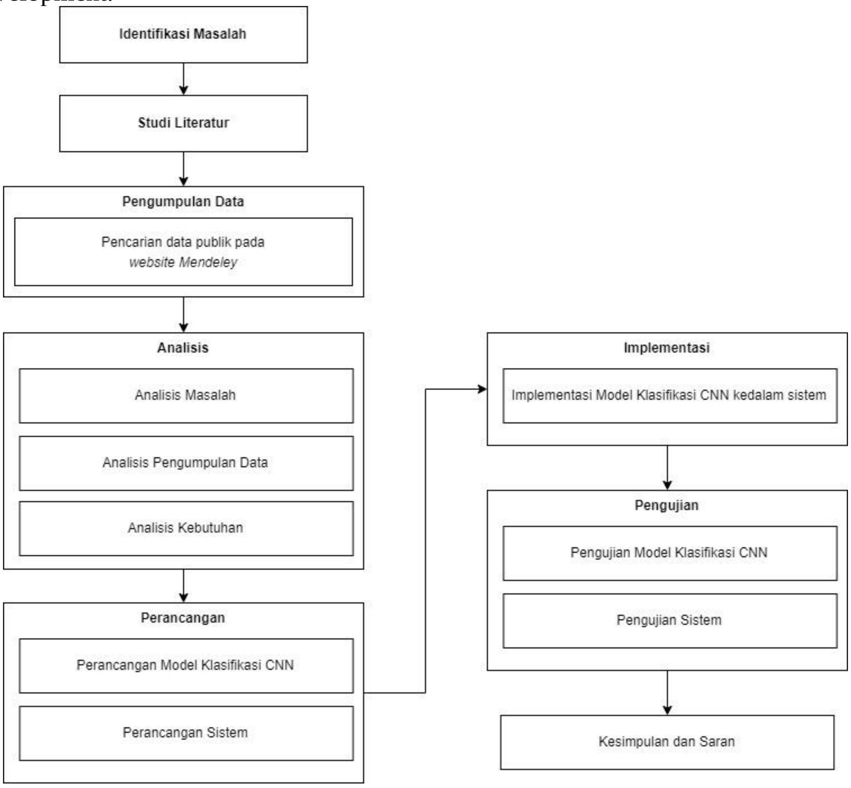

Figure 1. Research Stages

a. Identification Problem

Indonesia has an abundant wealth of herbal plants, with around 30,000 types of herbal plants, but only 1,200 types are used as traditional medicine ingredients. One of the main obstacles is the lack of public knowledge in recognizing herbal plants, especially the leaves that are often used as raw materials for medicine. Herbal plant leaves often look similar so that they are difficult to distinguish manually without special skills. As a result, the use of herbal plants for traditional medicine is often hampered by the manual identification process which is time-consuming, inaccurate, and inefficient.

b. Literature Study

The initial stage in this research is to conduct a literature study or research by collecting relevant information and data related to the problems, objectives, and methods to be applied. Study literature is activity with do search And collection of library data that supports the research to be carried out. Literature studies can be in the form of existing books, articles, journals and final reports. in relation to the research title.

c. Data collection

At the data collection stage, this study uses relevant datasets to support the herbal plant leaf classification process. The dataset used is public data taken from the Mendeley website with the title " *Indonesian Herb Leaf Dataset* 3500". The dataset contains 3,500 leaf images divided into 10 different classes. This data will be the basis for building an automatic herbal plant leaf image classification model.

d. Analysis

At the analysis stage, analysis is carried out on matters relating to the system. The analyses carried out are : problems and data collection .

e. Design

This stage is the stage carried out to design the system to be built. At this stage there are two types of design, namely , CNN classification model design and design system .

f. Implementation

The implementation stage implements the system design using a program so that the final result is a system or application in accordance with the design that has been made previously. The programming language used in the development of herbal leaf plant classification applications using CNN *transfer learning* is *python* and *JavaScript. The database* used for data storage uses MySQL and utilizes *the react native framework* to build applications on the *android platform.*

g. Testing

The testing stage is a trial stage of the system that has been built to ensure that the system is in accordance with expectations. At this stage there are 2 types of testing, namely , CNN classification model testing and testing. system

h. Conclusion and Suggestions

After carrying out all stages of research, the final stage is to draw conclusions from the research results and provide suggestions for further research development.

2.2 *Transfer Learning* Method

Transfer learning is a method of pre-trained *networks and a starting point for learning subsequent or new tasks.* **Invalid source specified.**. In *transfer learning, there is a* deeper *pooling* and *convolution* layer architecture compared to the regular CNN architecture. Therefore, *transfer learning* can produce better image information and more texture extraction **Invalid source specified.**. In this study , we will using transfer learning MobileNetV2 and InceptionV3. The following is description related both transfer learning .

a. MobileNetV2

MobileNetV2 is the first development of the MobileNet architecture. The difference with MobileNetV1 lies in the use of inverted residual blocks and linear bottlenecks. At the bottleneck, there are inputs and outputs that connect the model, while the inner layer functions to summarize the model's capabilities by converting inputs from lower levels, such as pixels, into higher-level descriptors. [11] . MobileNetV2 has 2 *convolutional blocks* , namely *residual block* from step 1 and step 2 for the purpose *downsizing.* Second block the contains 3 *layers.* Here is the description [12] .

1. First *layer* is 1x1 *convolutional layer* using ReLU6. ReLU6 is known for its robustness when used with low-precision calculations or computations. The use of 1x1 *convolutional layer* is to expand low-dimensional input feature maps to a higher-dimensional space suitable for non-linear activation.
2. Second *layer* is *convolutional layer* based depth using a 3x3 kernel. This implements one convolution filter per channel *input* and perform light filtering . With thus , he reach *filtering* higher dimensional spatial tensors .
3. third *layer* is report 1x1 convolution without non- linearity . This is because if use ReLU again, then will become linear classifier on *non-zero* volume parts from the output domain .

b. InceptionV3

InceptionV3 is a CNN model that has 42 layers. which is considered more efficient in aspect depth architecture , and smaller *error rates* . InceptionV3 carries different systems , namely use system factoring *Convolutional layers become multi-layers* with The kernel size is relatively smaller . This architecture has more precise parameters. Because capable reduce mark the parameters with share its weight become a number of *multi-layer* [13] . InceptionV3 has Lots filter sizes such as 1 * 1, 3 * 3, 5 * 5 can be applied at the same level . The following in Figure 2 is architecture from InceptionV3.

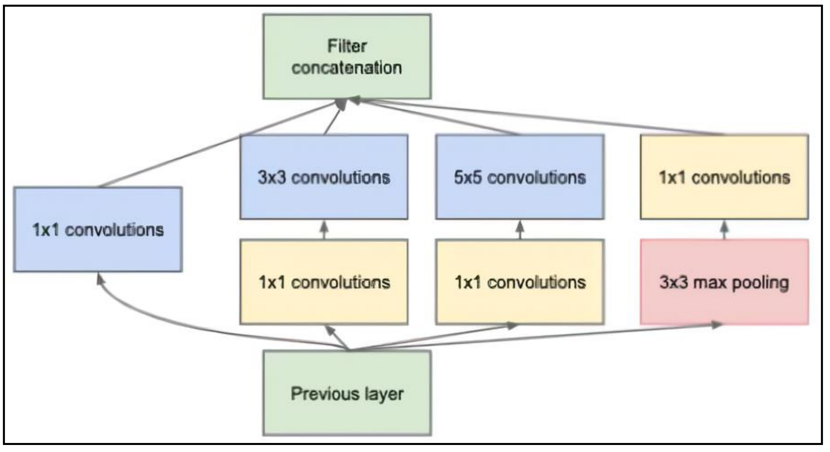

Figure 2 InceptionV3 Architecture

Based on Figure 2 of architecture the use 1x1 convolution to reduce burden computing before use the more expensive 3x3 and 5x5 convolutions . In addition to being used for subtraction , 1x1 convolutions also use linear activities that are improved to make them more efficient [14] .

2.3 *TensorFlow Libraries*

TensorFlow is a software library developed by the Google Brain team, originally designed for research and production at Google. It is widely used in machine learning development and makes it easier to build machine learning models [15] . *Tensorflow* use algebra computing and engineering optimization compilation to make it easier calculation significant amount from expression mathematics [16] .

3. RESULTS AND DISCUSSION

This chapter discusses the detection process . leaf herbal plants Convolutional Neural Network method with two Transfer Learning architectures, namely: MobileNetV2 and InceptionV3. From the two architectures, an accuracy test was then carried out by giving the same treatment between one architecture and another.

3 .1 Dataset

Dataset used in This research is public data taken from *Mendeley website* . The title of the dataset taken from *the Mendeley website* is " *Indonesian Herb Leaf Dataset 3500* ". The dataset consists of 10 classes with a total of 3500 photos or images. The 10 classes or categories of the dataset that will be used are as shown in Table 1.

Table 1 Herbal Plant Leaf Dataset Table

No	Latin Name	Indonesian Name	Picture
1	*Averrhoa bilimbi L.*	Starfruit	
2	*Psidium guajava*	Guava	
3	*Citrus aurantifolia*	Lime	
4	*Ocimum basilicum L.*	Basil	
5	*Aloe vera Lin*	Aloe vera	
6	*Artocarpus heterophyllus*	Jackfruit	
7	*Pandanus amaryllifolius Roxb .*	Pandanus	
8	*Carica papaya L*	Pawpaw	
9	*Apium graveolens L.*	Celery	
10	*Piper betle.L*	Betel	

3.2 Preprocessing

In the preprocessing process, four main stages are carried out to prepare the dataset into data that is ready to be processed using the *Transfer Learning approach* on the CNN algorithm. The four main processes include: the following.

1. Load Data

data loading step is carried out as an initial *preprocessing stage* that functions to read and load the dataset used for the herbal plant leaf image classification process in this application. The program code that is run to *load the data* is as shown in Table 2.

Table 2. Load Data

Program code (*python*)

dataset_path = PATH + 'Indonesian Herb Leaf Dataset 3500/'
 	
image_path = []
label = []
for x, i in enumerate(os.listdir (dataset_path)):
    new_data_path = dataset_path + str(i)
for filename in os.listdir (new_data_path):
        source_path = os.path .join (new_data_path , filename)
        image_path.append (source_path)
        label.append (new_data_path.split ('/')[-1])
df = pd.DataFrame ({' image_path ': image_path ,
'label': label
})
df

After *loading* the data using the code in Table 2 , the data calculation process for each class in the dataset is continued. The amount of data is shown in graphical form using the program code in Table 3 .

Table 3. Total Data for Each Class

Program code (*python*)
df.label .value_counts ()

The total data from each class in the dataset is 350 data, while the number of classes in the dataset used is 10 different classes, namely Starfruit, Guava, Lime, Basil, Aloe Vera, Jackfruit, Pandan, Papaya, Celery, and Betel as shown in Table 5.

Table 4. Total Data Display for Each Class

Label		
Starfruit 350		
Guava Seeds		350
Lime 350		
Basil	350	
Tongue Crocodile	350	
Jackfruit	350	
Pandanus	350	
Pawpaw	350	
Celery	350	
Betel	350	

2. Data Augmentation

The augmentation process is carried out at *the preprocessing stage* with the aim of increasing the variation of the dataset by making copies of modified herbal plant leaf images. The application of augmentation techniques used to increase the variation of the dataset consists of three types, namely *brightness* which is used to change the variation of brightness levels in the range between 0 (very dark) to 2 (very bright), *rotation* which is used to rotate the image randomly to a maximum angle of approximately 180 degrees, and the last technique applied is *vertical flip* used to flip the image vertically. The three augmentation techniques used are implemented using the program code in Table 5 below.

Table 5. Data Augmentation

Program code (*python*)

maximum_total_data = 500 # Set maximum of (original data + augmented data)
 	
# Define TensorFlow ImageDataGenerators with arrangement augmentation separated 	
brightness_datagen = tf.keras .preprocessing.image.ImageDataGenerator(brightness_range=[0,2])
rotation_datagen = tf.keras .preprocessing.image.ImageDataGenerator(rotation_range=180)
vertical_flip_datagen = tf.keras .preprocessing.image.ImageDataGenerator(vertical_flip=True)

After the augmentation process is carried out, the variation of data in the dataset will change . Because after the augmentation process, the variation of data in each class increases, so there is a change in the amount of data in each class. the amount of data in each class becomes 500 data which was previously only 350 so that the total data in the entire dataset after the augmentation process is 5000 data.

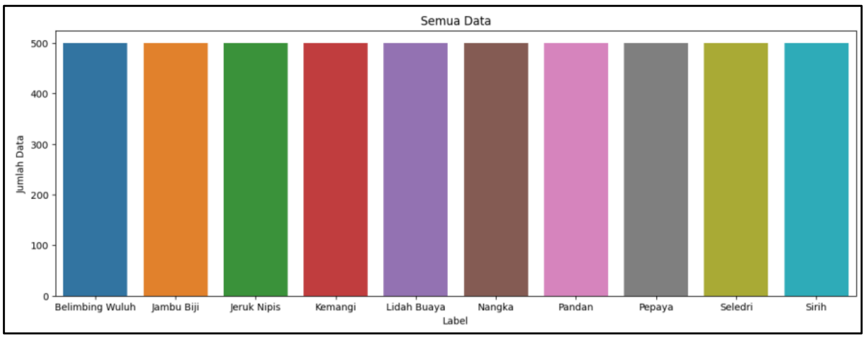

Figure 3. Total Data After Augmentation Process

3. Split Data

data split process is carried out to divide the dataset into three parts, namely *training data*, *testing* data , and *validation data*. *Training* data is used as model training data, *testing data* is used as test data on the model, and *validation data* is used to validate the data. The data division ratio applied is 80:20. This is intended by dividing the dataset into 80% *training data from the total amount*, 10% *testing data, and 10% validation data* , so that the amount of *training data* is 4000 data, 500 *testing data* , and 500 validation data. The program code applied for the data division process (*split data*) is as in Table 6 .

Table 6. Split Data

Program code (*python*)
train_df , test_df = train_test_ split (df_all_data , test_size =0.2, random_state =42) test_df , val_df = train_test_ split (test_df , test_size =0.5, random_state =42) train_df = train_ df.reset _index (drop=True) val_df = val_ df.reset _index (drop=True) test_df = test_df.reset_index (drop=True) print('Total training data :', train_df.shape [0]) print('Total validation data :', val_df.shape [0]) print('Total testing data :', test_df.shape [0])

Total training data: 4000

Total validation data : 500

Total testing data : 500

4. Export Dataset

proprocessing process carried out in this study for the application of herbal plant leaf image classification is to *export* the dataset into a file with the csv file type, using the program code in Table 7 .

Table 7. Export Dataset

Program code (*python*)
train_csv_path = os.path .join (PATH, 'train.csv') train_df.to_ csv (train_csv_path , index=False) validatoin_csv_path = os.path .join (PATH, 'validation.csv') val_df.to_ csv (validatoin_csv_path , index=False) test_csv_path = os.path .join (PATH, 'test.csv') test_df.to_ csv (test_csv_path , index=False)

3.3 Testing *Confusion Matrix*

System testing using the *confusion matrix method* aims to test the accuracy or precision of the system in classifying herbal plant leaves using a model that has been built and trained previously. The program code used for the system testing process by applying the *confusion matrix method* can be seen in Table 8.

Table 8. Confusion Matrix

Program code (*python*)
Confusion_matrix = confusion_matrix (test_generator.classes , y_pred) class_label = test_generator.class_ indices.keys () df_confusion = pd.DataFrame (Confusion_matrix , index = class_label , columns = class_label) sns.heatmap (df_confusion , annot = True, fmt = "d", cmap = plt.cm.Blues) plt.title ('Confusion Matrix') plt.xlabel ('Prediction Label') plt.ylabel ('Actual Label') plt.show () _accuracy = round(accuracy_score (test_generator.classes , y_pred)*100, 2) _ precision = round(precision_score (test_generator.classes , y_pred , average='weighted')*100, 2) _recall = round(recall_score (test_generator.classes , y_pred , average='weighted')*100, 2) _ fscore = round(f1_score(test_generator.classes , y_pred , average='weighted')*100, 2) print('Accuracy :', _accuracy, '%') print(' Precision :', _ precision , '%') print('Recall :', _recall, '%') print('F-Score :', _ fscore , '%')

1. *InceptionV3 Confusion Matrix* Test Results

From the test results, the evaluation values are grouped in the form of percentages consisting of four types, namely *accuracy, precision, recall,* and *f1-score* . Test results system use *InceptionV3* can be seen in Figure 4.

```
Accuracy    : 97.2 %
Precission  : 97.24 %
Recall      : 97.2 %
F-Score     : 97.2 %
```

Figure 4. InceptionV3 *Evaluation* Results

The evaluation results shown in Figure 4 with an average evaluation value as high as 97.2% indicate that the system is very accurate in predicting herbal plant leaves using *InceptionV3.*

2. *MobileNetV2 Confusion Matrix* Test Results

From the test results, the evaluation values are grouped in the form of percentages consisting of four types, namely *accuracy, precision, recall,* and *f1-score* . Test results system use *MobileNetV2* can be seen in Figure 5.

```
Accuracy   : 96.6 %
Precission : 96.73 %
Recall     : 96.6 %
F-Score    : 96.6 %
```

Figure 5. MobileNetV2 *Evaluation* Results

The evaluation results shown by **Error! Reference source not found.**with an average evaluation value as high as 96.63% indicate that the system is very accurate in predicting herbal plant leaves using *MobileNetV3.*

4. CLOSING

This study successfully implemented transfer learning with CNN architecture, namely InceptionV3 and MobileNetV2, for herbal plant leaf image classification. The system consists of a website application for admins to manage data and an Android application for users for direct classification. Testing showed that InceptionV3 had the highest accuracy of 97.2%, slightly superior to MobileNetV2 which reached 96.63%, proving the effectiveness of both architectures in detecting herbal leaf types.

BIBLIOGRAPHY

[1] A. Arifin, J. Hendyli, and DE Herwindiati, “Classification of Herbal Medicinal Plants Using Support Vector Machine Method,” *Comput. J. Comput. Sci. Inf. Syst.* , vol. 5, no. 1, p. 25, 2021, doi: 10.24912/computatio.v1i1.12811.

[2] M. Meiriyama, S. Devella, and SM Adelfi, “Herbal Leaf Classification Based on Shape and Texture Features Using KNN,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)* , vol. 9, no. 3, pp. 2573–2584, 2022, doi: 10.35957/jatisi.v9i3.2974.

[3] A. Herdiansah, RI Borman, D. Nurnaningsih, AAJ Sinlae, and RR Al Hakim, “Herbal Leaf Image Classification Using Backpropagation Neural Networks Based on Shape Feature Extraction,” *JURIKOM (Jurnal Ris. Komputer)* , vol. 9, no. 2, p. 388, 2022, doi: 10.30865/jurikom.v9i2.4066.

[4] S. Ratna, “Digital Image Processing and Histogram with Phyton and Phycharm Text Editor,” *Technol. J. Ilm.* , vol. 11, no. 3, p. 181, 2020, doi: 10.31602/tji.v11i3.3294.

[5] OV Putra, MZ Mustaqim, and D. Muriatmoko, “Transfer Learning for Rice Disease and Pest Classification Using MobileNetV2,” *Techno.Com* , vol. 22, no. 3, pp. 562–575, 2023, doi: 10.33633/tc.v22i3.8516.

[6] D. Martomanggolo, “Comparison of Convolutional Neural Network on Transfer Learning Method to Classify White Blood Cells,” *Ultim. J. Tech. Inform.* , vol. 13, no. 1, p. 51, 2021.

[7] Abdul Jalil Rozaqi, MR Arief, and A. Sunyoto, "Implementation of Transfer Learning in the Convolutional Neural Network Algorithm for Identification of Potato Leaf Disease," *Procedia Eng. Life Sci.* , vol. 1, no. 1, 2021, doi: 10.21070/pels.v1i1.820.

[8] NI Widiastuti, “Convolution Neural Network for Text Mining and Natural Language Processing,” *IOP Conf. Ser. Mater. Sci. Eng.* , vol. 662, no. 5, 2019, doi: 10.1088/1757-899X/662/5/052010.

[9] O. Saputra, DI Mulyana, and MB Yel, “Implementation of Convolutional Neural Network (CNN) Algorithm for Traditional Weapon Classification in Central Java Using Transfer Learning Method,” *J. SISKOM-KB (Computational Systems and Artificial Intelligence)* , vol. 5, no. 2, pp. 45–52, 2022, doi: 10.47970/siskom-kb.v5i2.282.

[10] MI Fathur Rozi, NO Adiwijaya, and DI Swasono, “Identification of the Performance of Transfer Learning Architecture Vgg16, Resnet-50, and Inception-V3 in Classifying Tomato Leaf Disease Images,” *J. Ris. Electrical Engineering* , vol. 5, no. 2, p. 145, 2023, doi: 10.30595/jrre.v5i2.18050.

[11] F. Marpaung, N. Khairina, R. Muliono, M. Muhathir, and S. Susilawati, “Classification of Ready-to-Harvest Tea Leaves Using Convolutional Neural Network Architecture Mobilenetv2,” *J. Teknoinfo* , vol. 18, no. 1, pp. 215–225, 2024, [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/article/view/3435

[12] ESMadhan, Arihant Kaushik, and Rohit Raju, “An Intelligent Dog Breed Recognition System Using Deep Learning,” *Int. J. Data Informatics Intel. Comput.* , vol. 1, no. 1, pp. 39–52, 2022, doi: 10.59461/ijdiic.v1i1.12.

[13] A. Varshney, A. Katiyar, AK Singh, and SS Chauhan, “Dog Breed Classification Using Deep Learning,” *2021 Int. Conf. Intel. Technol. CONIT 2021* , no. August, 2021, doi: 10.1109/CONIT51480.2021.9498338.

[14] A. Andrew and H. Santoso, “Compare VGG19, ResNet50, Inception-V3 for Review Food Ratings,” *SinkrOn* , vol. 7, no. 2, pp. 845–494, 2022, doi: 10.33395/sinkron.v7i2.11383.

[15] W. Saputra and YD Prabowo, “Development of Image Classification Application Using Tensorflow Library Implementing Convolutional Neural Network Algorithm Case Study: Photo Gallery of Shoot Fellowship Church Worship Activities,” *KALBISIANA J. Mhs. Inst. Technol. and Business Kalbis* , vol. 8, no. 3, pp. 2892–2901, 2022.

[16] M. Ihsan, RK Niswatin, and D. Swanjaya, “Facial Expression Detection Using Tensorflow,” *Joutica* , vol. 6, no. 1, p. 428, 2021, doi: 10.30736/jti.v6i1.554.