JUSIKOM PRIMA (Jurnal Sistem Informasi dan Ilmu Komputer Prima)
Vol. 8 No. 1, August 2024 E-ISSN:2580-2879

Herbal Leaf Image Classification Using Convolutional Neural Network (CNN)

Putra Edi Mujahid[*1] , Rosianni Manik[2], Junpri Sardodo Simbolon[3], Maria Riska Ratna Sari Sinaga[4],Siti Aisyah[5],Marlince Nababan[6]*
[1,2,3,4,5,6]*Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Prima Idonesia* [1]putraedimujahid@unprimdn.ac.id

ABSTRACT

This research delves into the application of Convolutional Neural Networks (CNNs) to address the complexities of identifying herbal leaf species in Indonesia, often challenging due to thevast variations in shape, color, and texture. Utilizing a dataset of herbal leaf images acquired using the Bing Downloader Scrapping technique, a CNN model was trained to classify various plant varieties with a remarkable accuracy rate of 92.66%. Additionally, the analysis of low loss values indicates that the model not only effectively maps the intricate features of each image to the correct category but also efficiently reduces error rates. These findings offer a significant contribution to the context of herbal medicine development and biodiversity conservation, opening up avenues for technological integration in efforts to preserve Indonesia's natural and cultural resources.

Keywords: *Convolutional Neural Network* (CNN), *Bing Downloader, Herbal Plants, ImageClassification.*

INTRODUCTION

Indonesia's vast tropical rainforests harbor an extraordinary biodiversity, including a wealth of herbal plants. Local wisdom in utilizing these plants for medicinal purposes has long been practiced by Indonesian communities [1][2]. Leaves, rich in vitamins, minerals, antioxidants, and other beneficial substances, are widely used in herbal remedies. Herbal medicines typically originate from plants containing phytochemical compounds that play a crucial role in maintaining and promoting overall health [3]. According to [4], Indonesia ranks second in the world after Brazil in terms of plant diversity. Out of approximately 30,000 plant species in Indonesia, research indicates that 6,000 have the potential to serve as medicines. Other sources estimate the number of plant species in Indonesia to exceed 7,000, with around 1,000 already utilized for disease prevention and treatment. Demonstrating its commitment, Indonesia isamong 25 countries that have developed and implemented natural medicine policies.

The abundance of herbal leaf varieties poses challenges for direct identification. Leaf characteristics such as shape, color, and texture serve as crucial data for classification. However, current manual classification methods involving visual observation and individual assumptions have limitations. The process is time-consuming and prone to errors [5]. Therefore, a technology capable of accurately classifying herbal leaf types based on the patterns of shape, color, and texture variations is essential [6][7].

K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest are methods widely used in image processing, achieving relatively good accuracy rates. However, among these methods, Convolutional Neural Network (CNN) has emerged as the most prevalent and effective [8].

Numerous researchers have ventured into herbal leaf classification. [9] employed the Backpropagation Neural Network (BNN) algorithm for herbal leaf classification, extracting leaf shape features using metric and eccentricity parameters to aid the BNN algorithm in classification. The segmentation results were optimized using morphological operations toenhance accuracy, achieving an accuracy of 88.75% during testing. This highlights the needfor further method development, leading to the research titled "Herbal Leaf ImageClassification Using Convolutional Neural Network (CNN)"..

LITERATURE REVIEW

The literature review represents the theoretical core of an article. In this section, we will discussthe purpose of a literature review. We will also consider how one should go about to find appropriate literature on which to base a literature review and how this information should be managed. Finally, we will answer four questions that first-time researchers often battle with when compiling a literature review.

These questions are: which aspects should I include in a literature review?; how should I go about synthesizing information in a literature review?; how should I structure a literature review? what writing style should I use when compiling a literature review?

The purpose of a literature review is to “look again” (re + view) at what other researchers

have done regarding a specific topic (Leedy & Ormrod 2005:70). A literature review is a means toan end, namely to provide background to and serve as motivation for the objectives and hypotheses that guide your own research (Perry et al. 2003:660)

A good literature review does not merely summarise relevant previous research. In the literature review, the researcher critically evaluates, re-organizes and synthesizes the work of others (Leedy & Ormrod, 2005:84). In a sense, compiling a literature review is like making a smoothie or fruit shake: The end product is a condensed mix that differs totally in appearance from the individual ingredients used as inputs. The key to a successful literature review lies in your ability to "digest" information from different sources, critically evaluate it and resent your conclusions in a concise, logical and reader-friendly" manner.

First-time researchers often naively believe everything they read or are scared to criticize the work of others. However, academic research is all about critical inquiry! It is, therefore, extremely important that you critically evaluate the material that you read. Do you agree with the arguments and conclusions of other researchers? If you disagree, why? Can you identify contradictory arguments or findings? How could one explain these contradictions? Do the findings of previous studies apply in all contexts or are the findings context-specific? What are the criticisms against the conceptual models or measurement approaches discussed in the literature? Which limitations should be considered when interpreting the results of previous research?

You have to carefully read the most recent available literature to identify specific gaps, inconsistencies and/or controversies that may form the basis of your own research. Always show that you have considered an issue from several angles and that you are aware of the arguments for and against a specific point of view. Many researchers in services marketing, for example, use the SERVQUAL measurement scale without considering existing criticisms against it.

To compile a proper literature review, one has to overcome three specific challenges, namely: finding appropriate literature on a specific topic, managing the information, and presenting a logical, synthesized, and reader-friendly review of the current knowledge relating to a specific topic. Consider the following search strategies: Blackwell Synergy; Proquest Data

Basis; EBSCOhost (Business Source Premier and Business Source Premier); Emerald; Taylor and Francis; Infotrac; Wiley Interscience; and others open access journal using Google Scholar. To view information about the "literature review" more fully, please visit the link

METHODS

Convolutional Neural Networks (CNNs) stand as a cornerstone of modern artificial intelligence, revolutionizing the realm of image and video recognition. Inspired by the intricate workings of the human brain, CNNs excel at extracting meaningful patterns and features from visual data, paving the way for groundbreaking advancements in computer vision.

The intricate structure of CNNs comprises several key components that work in harmony to achieve remarkable recognition capabilities:

a. Input: The journey of image recognition begins with the input layer, receiving the imageor video data that the CNN aims to decipher.

b. Feature Extraction: At the heart of CNNs lies the feature extraction stage, where the network embarks on a quest to uncover the hidden patterns and characteristics that define the input data.

c. Convolutional Layer: The convolutional layer serves as the CNN's workhorse, employing filters to meticulously scan the input data, akin to applying a stamp to extractprominent visual features.

d. Activation Function (ReLU): The ReLU (Rectified Linear Unit) activation function acts as a gatekeeper, ensuring that only relevant information passes through the network, amplifying the extracted features while suppressing insignificant ones.

e. Pooling: The pooling layer plays a crucial role in dimensionality reduction, summarizing the extracted features by taking the average or maximum value from smaller regions of the image. This process reduces computational complexity and

enhances the network's robustness to minor variations in the input data.

f. Flattening: Once the feature extraction stage has successfully distilled the essence of the input data, the flattening layer transforms the multi-dimensional feature map into a one-dimensional vector, preparing it for the classification stage.

g. Classification: The classification stage marks the culmination of the CNN's journey, where the extracted features are finally utilized to categorize the input data. This intricate process involves a fully connected layer, a neural network that analyzes the flattened feature vector and assigns the input data to the most appropriate category.Output: The output layer proudly unveils the CNN's verdict, presenting the classification result, which represents the category to which the input image or video belongs.

CNNs work in stages, where the results from one layer become input for the next layer. This allows CNN to learn complex patterns and improve accuracy [11] [12].

CNN is a computational model that is optimized for processing images. The advantage of CNN lies in its ability to handle spatial data, such as two-dimensional images, through convolution operations. This operation involves applying a filter to the image, similar to a stamp, to extract important features. Convolutional layers act as the core of a CNN, performing mathematical calculations to identify hidden patterns within images.

CNN functions are not only limited to feature extraction. This network has several other layers that play important roles. The pooling layer plays a role in data reduction by taking the average or maximum value from parts of the image. The fully connected layer acts as a final classifier, determining image categories based on feature extraction results.

In this research, the specific CNN architecture will be discussed further, as shown in Figure 2. This architecture is like a blueprint, determining the settings and settings of the various layers, so that the CNN can work optimally to achieve the research objectives.

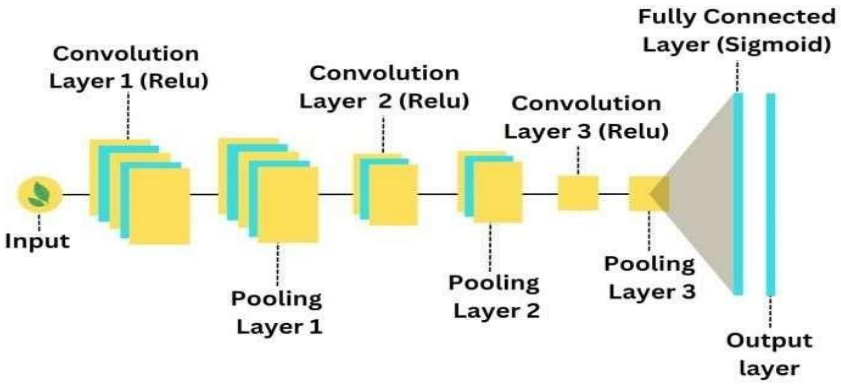


Figure 1 CNN Architecture [13]

RESULTS

Problem Analysis

Herbal leaf image classification using Convolutional Neural Network (CNN) is promising research in increasing the efficiency and accuracy of herbal plant identification. CNN is able

to learn complex patterns in leaf images and produce accurate classification compared to traditional methods which are manual and prone to errors

Data Processing

The data processing process is carried out in 5 stages which can be seen in Figure 2 below:

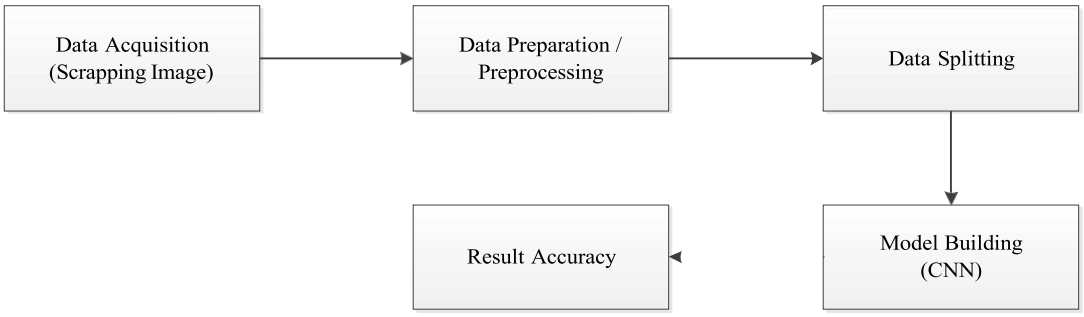


Figure 2 Data Processing Flow

Data Acquisition

Data Acquisition(Scrapping Image) where herbal leaf images are collected from the Bing Browser using the Bing Downloader Library, downloaded, and annotated with the correct species labels. Data quality and diversity were ensured to increase model reliability. The scrapping process is carried out with 10 different types of plants such as guava leaves, curry leaves, basil leaves, turmeric leaves, mint leaves, papaya leaves, betel leaves, soursop leaves, aloe vera and green tea which we can see in figure 3 below.

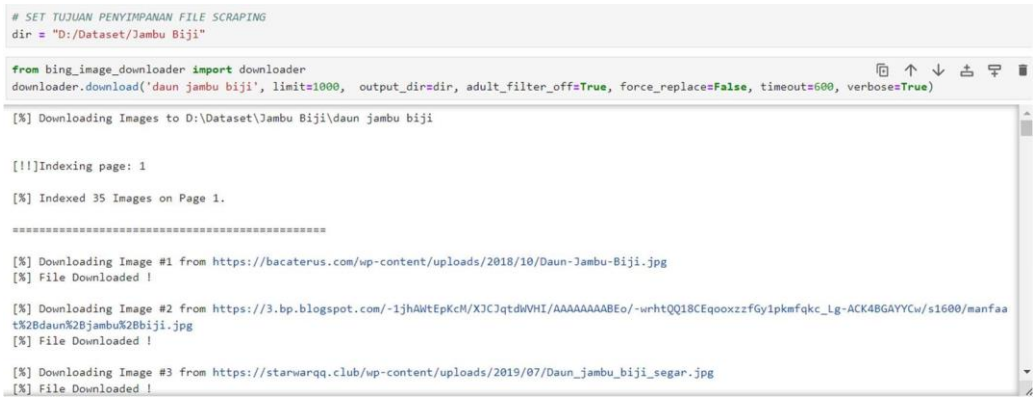


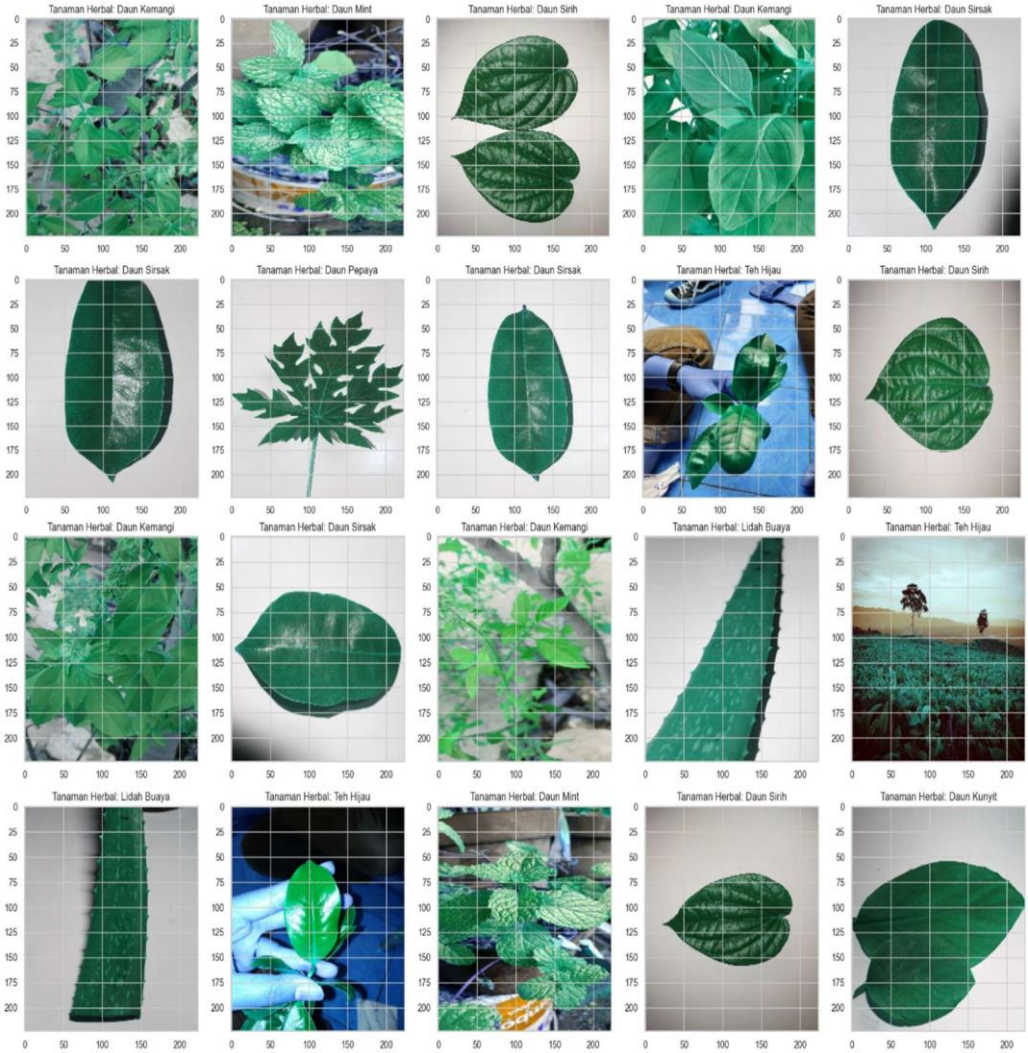
Figure 3 ***Scrapping Data Bing Downloader***

The scrapping process carried out obtained 4400 images with details of 80 images of guava leaves, 160 images of curry leaves, 240 images of basil leaves, 320 images of turmeric leaves,

400 images of mint leaves, 480 images of papaya leaves, 560 images of betel leaves, 640images of soursop leaves. , Aloe Vera 720 images, and Green Tea 800 images

```
fig,ax=plt.subplots(5,5)
fig.set_size_inches(20,20)
for i in range(5):
    for j in range (5):
        l=rn.randint(0,len(Z))
        ax[i,j].imshow(X[l])
        ax[i,j].set_title('Tanaman Herbal: '+Z[l])

plt.tight_layout()
```



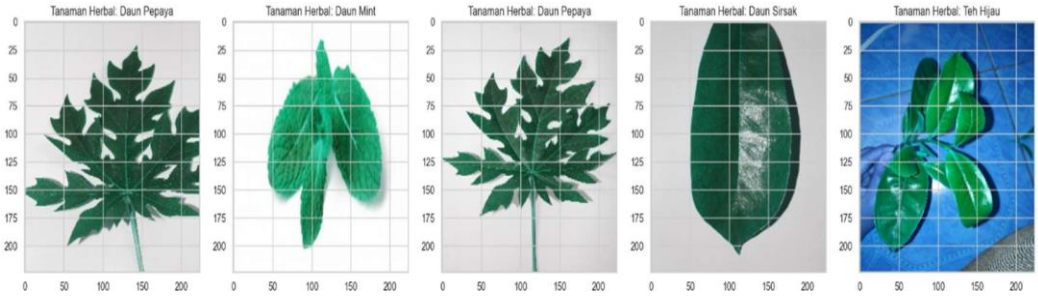


Figure 4 Image Dataset

Data Preperation (Preprocessing)

Data preprocessing is the next step, where the steps taken are reading images from the specifieddirectory, resizing the image to the specified dimensions, assigning image labels based on the directory name, and adding the processed images and labels to the X and Y lists.

Resize Image

The image obtained from the previous stage will be changed to the default size of 224x 224 to avoid overfitting and underfitting.

Add processed images and labels to the X and Y lists

This process is carried out so that all images obtained from scrapping can be directly used as input for training an image classification model.

The preprocessing process can be seen in Figure 5 below

```
X=[]
Z=[]
IMG_SIZE=224
Daun_Jambu_Biji='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Jambu Biji'
Daun_Kari='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Kari'
Daun_Kemangi='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Kemangi'
Daun_Kunyit='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Kunyit'
Daun_Mint='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Mint'
Daun_Pepaya='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Pepaya'
Daun_Sirih='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Sirih'
Daun_Sirsak='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Daun Sirsak'
Lidah_Buaya='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Lidah Buaya'
Teh_Hijau='C:/Users/stive/Downloads/Compressed/DATASET TANAMAN HERBAL/Data Training/Teh Hijau'
```

```
def assign_label(img,tanaman_herbal):
    return tanaman_herbal
```

Figure 5 Preprocessing Steps

Data Splitting

The data sharing process is carried out by dividing the training data and testing data to train

the model which will be carried out in the next stage. The training set is used to train the CNN model, validation is used to evaluate the performance of the model during training and prevents overfitting, test is used to evaluate the final performance of the model after training. In the

research, the division is divided into 80:20 where 80% of the data is training data and 20% istesting data. We can see in Figure 6 below.

```
le=LabelEncoder()
Y=le.fit_transform(Z)
Y=to_categorical(Y,10)
X=np.array(X)
X=X/255
```

```
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.20,random_state=42)
```

Figure 6 Dataset Spiliting

Model Building

CNN (Convolutional Neural Networks) is an artificial neural network designed to process spatial data such as images. CNN works by extracting important features from images througha series of convolution and pooling operations. CNN is a very effective method for image processing with various applications. Despite some shortcomings, CNNs offer high accuracy and efficiency in image classification and other image processing tasks.

```
# # modelling starts using a CNN.

model = Sequential()
model.add(Conv2D(filters = 32, kernel_size = (17,17),padding = 'Same',activation ='relu', input_shape = (224,224,3)))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',activation ='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Conv2D(filters =96, kernel_size = (3,3),padding = 'Same',activation ='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Conv2D(filters = 96, kernel_size = (3,3),padding = 'Same',activation ='relu'))
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2)))

model.add(Flatten())
model.add(Dense(512))
model.add(Activation('relu'))
model.add(Dense(10, activation = "softmax"))
```

Figure 7 Model Building

Convolution Layer (Convolutional Layer)

a. filters=32: Number of filters used by the convolution layer. This filter is used to

extract features from the input image. In this model, the first convolution layer has 32 filters.

b. kernel_size=(17,17): Size of the convolution kernel. This kernel is like a window that moves across the input image to extract features. The larger the kernel size, the larger the image area considered for feature extraction.

c. padding='Same': Padding option that ensures the output of the convolution layer hasthe same spatial dimensions as the input. This is achieved by adding zero padding around the input image.

d. activation='relu': The activation function used on the output of the convolution layer. ReLU (Rectified Linear Unit) is a generalized activation function that introduces non- linearity into the model.

e. input_shape=(224,224,3): Determines the shape of the input image. In this model, it is assumed that the input image has a size of 224x224 pixels with 3 color channels (RGB).

f. This model has 4 convolution layers in total, each with:

- Second layer: 64 filters, kernel size (3.3)

- Third and fourth layers: 96 filters, kernel size (3.3)

```
datagen = ImageDataGenerator(
        featurewise_center=False,  # set input mean to 0 over the dataset
        samplewise_center=False,  # set each sample mean to 0
        featurewise_std_normalization=False,  # divide inputs by std of the dataset
        samplewise_std_normalization=False,  # divide each input by its std
        zca_whitening=False,  # apply ZCA whitening
        rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
        zoom_range = 0.1, # Randomly zoom image
        width_shift_range=0.2,  # randomly shift images horizontally (fraction of total width)
        height_shift_range=0.2,  # randomly shift images vertically (fraction of total height)
        horizontal_flip=True,  # randomly flip images
        vertical_flip=False)  # randomly flip images

datagen.fit(x_train)
```

```
model.compile(optimizer='Adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

Figure 8 Dataset Generator

a. featurewise_center and samplewise_center: These arguments are set to False

because they center pixel values (either for the entire dataset or for each image) around the meanvalue.

b. featurewise_std_normalization and samplewise_std_normalization: These arguments are also set to False because they normalize pixel values (either for the entire dataset or for each image) based on the standard deviation.
c. zca_whitening: This argument is set to False and disables ZCA whitening, a technique that can improve convergence for certain datasets.
d. rotation_range=10: This argument specifies the random rotation range (in degrees) to be applied to the image during augmentation. Here, it is set to 10 degrees, meaning the image can be rotated between -10 and 10 degrees.
e. zoom_range=0.1: This argument specifies the random zoom range applied to the image. Here, it is set to 0.1, allowing a maximum zoom in or out of 10%.
f. width_shift_range=0.2, height_shift_range=0.2: This argument controls the random horizontal and vertical shift range applied to the image (as a fraction of the total width or height). Here, they allow a maximum shift of 20% in either direction.horizontal_flip=True: This argument enables random horizontal flips in the image, which can help the model learn to be invariant to left-right variations.
g. vertical_flip=False: This argument disables random vertical flipping. You can enable it if horizontal reversal alone doesn't provide enough variation.
h. datagen.fit(x_train): This line calls the fit method on the datagen object. The fit method is used to calculate statistics (such as mean and standard deviation) based on theprovided x_train training data

Accuracy of Results

In presenting the accuracy results of the trained model, it can be seen in Figure 9 below:

```
batch=25
epoch=50
step = np.ceil(x_train.shape[0]/batch)
History = model.fit_generator(
    datagen.flow(x_train,y_train, batch_size=batch),
    epochs = epoch,
    validation_data = (x_test,y_test),
    verbose = 1,
    steps_per_epoch=step
)
```

Figure 9 Data Accuracy

Tuning Hyperparameters:

a. batch = 25: This line specifies the batch size, i.e. the number of training samples processed by the model in each iteration (step) during training. A batch size of 25 means25 images and their associated labels will be inserted into the model at once.

b. epoch = 50: This line specifies the number of training epochs. An epoch represents a complete iteration through the entire training dataset. Here, the model will be trainedfor 50 epochs.

Counting Steps per Epoch:

a. step = np.ceil(x_train.shape[0] / batch): This line calculates the number of steps (iterations) required to complete one epoch:

b. x_train.shape[0] gives the total number of samples in your training data (x_train).

c. batch is the previously defined batch size.

d. np.ceil() ensures we get the number of steps rounded up, because we want to process all training samples even if they don't fit into the entire batch. This is important to avoid skipping any training data.

Training the Model:

a. History = model.fit_generator(...): This line calls the fit_generator method on your CNN model (model) to start the training process.

b. datagen.flow(x_train, y_train, batch_size=batch): This argument provides the data generator (datagen) that you previously configured. The flow method generates a

batch of augmented training data (images and labels) from x_train and y_train with thespecified batch size.

c. epochs=epoch: This argument specifies the number of epochs (50) as previously specified.
d. validation_data=(x_test, y_test): This argument provides validation data (x_test and y_test) for evaluation during training. The model will periodically assess its performance on this separate validation set.
e. verbose=1: This argument sets the verbosity level to 1. The model will print training progress information during each epoch.
f. steps_per_epoch=step: This argument tells the fit_generator method how many steps (iterations) to perform in each epoch. This ensures that all training data is used for training in one epoch, taking into account the calculated step values.

After setting the configuration in machine learning above, the accuracy value obtained for 50 experimental epochs with a depth of 26 layers obtained the highest accuracy of 92.66% at the 49th epoch. We can see the resulting epoch value in Figure 10 below:

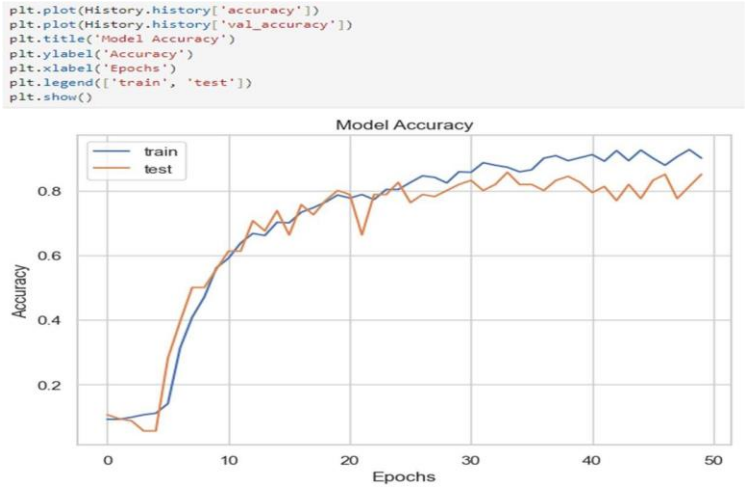


Figure 10 Accuracy Value

After experimenting 50 times with an accuracy of 92.66%, you also need to look at the model loss value. Model loss (or model loss function) is a metric used to measure how bad the model estimates are. It shows how far the model predictions deviate from the true labels (target values)for each data sample. The lower the model loss, the better the model is at

mapping inputs to the correct outputs. From Figure 11 it can be seen that in epoch 1 – 50 the value is getting lower.

```
plt.plot(History.history['loss'])
plt.plot(History.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'test'])
plt.show()
```

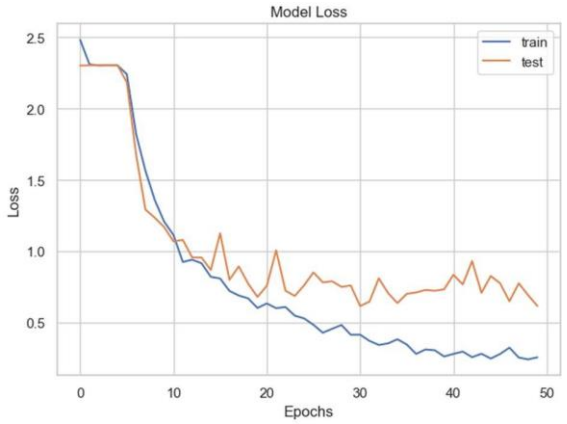


Figure 11 Accurate Loss Model

DISCUSSION

Suggestions for model development are Test other model architectures: Try more advanced CNN architectures, such as ResNet, Inception, or EfficientNet, which can improve model recall and efficiency.

CONCLUSION

Based on the experimental results, the drilled model shows excellent performance with an accuracy of 92.66%. This shows that the model is able to classify the data correctly with a high level of certainty.

A small model loss value further indicates that the model learns well and can map input to the correct output with a low error rate..

REFERENCES

Adela Regita Azzahra, "Klasifikasi Daun Herbal Menggunakan Metode CNN dan Naïve Bayes dengan Fitur GLCM," Indones. J. Comput. Sci., vol. 12, no. 4, 2023, doi: 10.33022/ijcs.v12i4.3362.

A. K. S. Yuda and S. Ahmad, "Implementasi Prediksi Tanaman Herbal Menggunakan

Algoritma Convolutional Neural Network Berbasis Android.," Reputasi J. Rekayasa Perangkat Lunak, vol. 4, no. 2, pp. 84–88, 2023, doi: 10.31294/reputasi.v4i2.2403.

R. J. Rumandan, R. Nuraini, N. Sadikin, and Y. Rahmanto, "Klasifikasi Citra Jenis Daun Berkhasiat Obat Menggunakan Algoritma Jaringan Syaraf Tiruan Extreme Learning Machine," J. Comput. Syst. Informatics, vol. 4, no. 1, pp. 145–154, 2022, doi: 10.47065/josyc.v4i1.2586.

"Direktorat Jenderal Pelayanan Kesehatan." Accessed: Feb. 07, 2024. [Online]. Available: https://yankes.kemkes.go.id/view_artikel/13/perkembangan-obat-dan-pengobatan-tradisional-dalam-kesehatan-masyarakat-dan- pemanfaatannya-di-rumah-sakit

A. M. Atha and E. Zuliarso, "Deteksi Tanaman Herbal Khusus Untuk Penyakit KulitDan Penyakit Rambut Menggunakan ConvolutionalNeural Network (CNN) Dan Tensorflow," J. JUPITER, vol. 4 (2), pp. 1–10, 2022.

I. N. Purnama, "Herbal Plant Detection Based on Leaves Image Using Convolutional Neural Network With Mobile Net Architecture," JITK (Jurnal Ilmu Pengetah. danTeknol. Komputer), vol. 6, no. 1, pp. 27–32, 2020, doi: 10.33480/jitk.v6i1.1400.

P. Purwanto and S. Sumardi, "Perancangan Klasifikasi Tanaman Herbal Menggunakan Transfer Learning Pada Algoritma Convolutional Neural Network (CNN)," J. Ilm. Infokam, vol. 18, no. 2, pp. 105–118, 2022, doi: 10.53845/infokam.v18i2.328.

M. H. Ahmad, F. M. Hana, T. G. Pratama, and H. Aulida, "Klasifikasi Empat Jenis Daun Herbal Menggunakan Metode Convolutional Neural Network," J. Ilmu Komput. dan Mat., vol. 4, no. 2, pp. 69–76, 2023.

A. Herdiansah, R. I. Borman, D. Nurnaningsih, A. A. J. Sinlae, and R. R. Al Hakim, "Klasifikasi Citra Daun Herbal Dengan Menggunakan Backpropagation Neural Networks Berdasarkan Ekstraksi Ciri Bentuk," JURIKOM (Jurnal Ris. Komputer), vol. 9, no. 2, p. 388, 2022, doi: 10.30865/jurikom.v9i2.4066.

H. Fauzi Jessar, A. Toto Wibowo, and E. Rachmawati, "Klasifikasi Genus Tanaman Sukulen Menggunakan Convolutional Neural Network," e-Proceeding Eng., vol. 8, no. 2, p. 3180, 2021.

S. P. Backar, P. Purnawansyah, H. Darwis, and W. Astuti, "Hybrid Fourier Descriptor Naïve Bayes dan CNN pada Klasifikasi Daun Herbal," J. Inform. J. Pengemb. IT, vol. 8, no. 2, pp. 126–133, 2023, doi: 10.30591/jpit.v8i2.5186.

Haryono, Khairul Anam, and Azmi Saleh, "Autentikasi Daun Herbal Menggunakan Convolutional Neural Network dan Raspberry Pi," J. Nas. Tek. Elektro dan Teknol. Inf., vol. 9, no. 3, pp. 278–286, 2020, doi: 10.22146/.v9i3.302.

A. R. Rahmadani et al., "Klasifikasi Citra Digital Daun Herbal Menggunakan Support Vector Machine dan Convolutional Neural Network dengan Fitur Fourier Descriptor," vol. 16, no. 1, 2024.

E. S. Barus, J. E. Halim, and S. Yessica, "Comparative Analysis of Stroke Classification

Using the K-Nearest Neighbor Decision Tree, and Multilayer Perceptron Methods," J. Sist. Inf. dan Ilmu Komput. Prima(JUSIKOM PRIMA), vol. 7, no. 1, pp. 155–167, 2023, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v7i1.4083.

Y. Sitinjak, M. Nababan, and M. City, "LIVER DISEASE CLASSIFICATION ANALYSIS," vol. 7, no. 1, pp. 132–141, 2023.