P-ISSN: 2656-1743 | E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i3.373

JURNAL RISET INFORMATIKA
Vol. 7, No. 3. Juni 2025

Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

LONG BEAN LEAF DISEASE IDENTIFICATION SYSTEM BASED ON MOBILE USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

Muhammad Fadilah Nurjaman-1, Gina Purnama Insany-2, Imam Sanjaya-n

[1,2,3] Teknik Informatika / Fakultas Teknik, Komputer dan Desain
[1,2,3] Universitas Nusa Putra
muhammad.fadilah_ti21@nusaputra.ac.id [1], gina.purnama@nusaputra.ac.id [2],
imam.sanjaya@nusaputra.ac.id [3]

Abstract

Long beans (Vigna unguiculata subsp. sesquipedalis), have high nutritional value, besides long beans also have a significant role in the economy of farmers in Indonesia. However, the productivity of this plant is often hampered by various diseases that attack the leaves, which can result in a decrease in the quantity and quality of the harvest. This study has succeeded in developing a Convolutional Neural Network (CNN) model with the ResNet-50 architecture to identify six types of diseases in long bean leaves. The dataset used consists of 2,316 images, divided into training data (80%), validation (15%), and testing (5%). The ResNet-50 model, which consists of 50 layers, applies the transfer learning technique by not training the first 35 layers using a specific dataset, but utilizing weights from ImageNet. Training for 100 epochs produces high accuracy, namely 98.3% for training data, 98.4% for validation data, and 98.7% for testing data. Evaluation using Confusion Matrix, Precision, Recal and F1 Score shows very good performance without prediction errors. The final result of this research is a mobile-based software system that can diagnose diseases quickly and accurately, which can help farmers take appropriate action, and support sustainable agriculture in Indonesia.

Keywords: Long Bean, Plant Disease, Convolutional Neural Network (CNN), ResNet-50, Mobile

Abstrak

Kacang panjang (Vigna unguiculata subsp. sesquipedalis), memiliki nilai gizi yang tinggi, selain itu kacang panjang juga memiliki peran signifikan dalam perekonomian petani diindonesia. Namun, produktivitas tanaman ini seringkali terhambat oleh berbagai penyakit yang menyerang daun, yang dapat mengakibatkan penurunan kuantitas serta kualitas hasil panen. Penelitian ini telah berhasil mengembangkan model Convolutional Neural Network (CNN) dengan arsitektur ResNet-50 untuk mengidentifikasi enam jenis penyakit pada daun kacang panjang. Dataset yang digunakan terdiri dari 2.316 gambar, dibagi menjadi data pelatihan (80%), validasi (15%), dan pengujian (5%). Model ResNet-50, yang terdiri dari 50 layer, menerapkan teknik transfer learning dengan tidak melatih 35 layer pertama menggunakan dataset yang spesifik, melainkan memanfaatkan bobot dari ImageNet. Pelatihan selama 100 epoch menghasilkan akurasi tinggi, yaitu 98,3% untuk data pelatihan, 98,4% untuk data validasi, dan 98,7% untuk data pengujian. Evaluasi menggunakan confusion matrix, precision, recal dan f1 score menunjukkan performa yang sangat baik tanpa kesalahan prediksi. Hasil akhir dari penelitian ini berupa sebuah sistem perangkat lunak berbasis mobile yang dapat mendiagnosis penyakit dengan cepat dan akurat, yang dapat membantu petani mengambil tindakan tepat, serta mendukung pertanian berkelanjutan di Indonesia.

Kata kunci: Kacang Panjang, Penyakit tanaman, Convolutional Neural Network (CNN), ResNet-50, Mobile.

INTRODUCTION

Long beans (Vigna unguiculata subsp. sesquipedalis), which originate from Africa, grow widely throughout Asia, including Southeast, China, Europe, Oceania, and North (Azka & Sayekti, 2020). In addition to having high nutritional value, long beans also play a significant role in the economy of farmers in Indonesia. However, the productivity of this plant is often hampered by various pests and diseases that attack the leaves(Arafa et al., 2024). So that it can result in a decrease in the quantity and quality of the harvest.

Based on data from the Central Statistics Agency on crop and vegetable production in 2021-2023, long bean production in Indonesia tends to decrease from year to year. Production in 2021 was 383,685 tons, then in 2022 it became 360,871 tons, and decreased again in 2023 to 309,422 tons(Badan Pusat Statistik indonesia, 2024). One of the causes of this decline is due to diseases that attack the leaves of long bean plants, resulting in a decrease in the quality and quantity of crop production.

Early identification of diseases in long bean plants is very important to minimize the negative impacts that may occur (Kusuma et al., 2025). Therefore, a more modern and efficient approach is needed in detecting diseases in plants. One way to identify is to recognize the structural characteristics of the leaves, such as spots that appear on the surface of the leaves (Hidayati et al., 2020). To analyze these characteristics, image processing can be carried out using digital image processing techniques, which allow for more accurate and faster disease detection (J. V. P. Putra et al., 2023).

In recent years, the development of technology in the field of artificial intelligence and digital image processing has been increasingly rapid (Afandi & Kurnia, 2023). One of the most widely used methods in image processing is the Convolutional Neural Network (CNN) method (Peryanto et al., 2020). CNN can be used to analyze plant leaf images and identify disease symptoms with a high level of accuracy (Maysela & Rohma, 2024). One of the popular CNN architectures is ResNet, which is known to be able to overcome the vanishing gradient problem and increase accuracy in the classification process (Setyadi et al., 2024).

In previous studies, there were several studies that discussed the use of the Convolutional Neural Network (CNN) algorithm to diagnose plant diseases. One of them is a study on the classification of corn leaf diseases using the CNN method, which succeeded in achieving the highest accuracy of 98.3% (I. P. Putra et al., 2022). In a study on disease identification in potato leaf images using the Convolutional Neural Network (CNN) method, the results obtained showed that the accuracy for training data reached 93%, while the highest accuracy for validation data reached 99% (Lesmana et al., 2022). In a study that applied the Residual Network (ResNet) method for classifying diseases in wheat leaves, the Convolutional Neural Network (CNN) method successfully achieved an accuracy of 92.68% (Suprihanto et al., 2022). From these results, it can be concluded that the classification method using the Convolutional Neural Network (CNN) has proven to be quite effective in classifying or identifying diseases in plants.

The novelty of this study lies in several important aspects that differentiate it from previous studies. First, this study uses a new dataset specifically collected for long bean leaves, covering various types of diseases that commonly attack this plant, such as yellow mosaic, bean rust, bean angular leaf spot, powdery mildew, and leaf crinkle. This dataset has never been used before in the context of disease identification in long beans, thus providing a new contribution to the existing literature. Second, this study applies a special augmentation technique designed to increase data variation and strengthen the model in recognizing different disease symptoms, including rotation, cutting, and lighting changes that are tailored to local conditions in Indonesia. Third, this study also focuses on adaptation to local conditions in Indonesia, taking into account specific environmental factors and agricultural practices. The identified disease symptoms include yellow mosaic which is characterized by irregular yellow spots on the leaves, bean rust which is characterized by small white spots that enlarge and turn into floury brown, bean angular leaf spot which shows mottled leaves and blackish spots, powdery mildew which appears as a white layer like flour on the leaf surface, and leaf crinkle which is characterized by curved and wrinkled leaves. By understanding the symptoms of this disease, this study aims to provide more relevant and applicable solutions for farmers in Indonesia, so that it is expected to help increase the productivity and sustainability of agriculture in Indonesia.

RESEARCH METHODS

The research method applied in this study is an experimental method which is descriptive, quantitative, and evaluative. The Convolutional Neural Network (CNN) architecture uses the ResNet-50 model. The dataset used in this study comes from two dataset provider websites, namely Huggingface.co and Data.mendeley.com with a total of 1717 images divided into 6 different classes.

Types of research

The research method applied in this study is an experimental method that is descriptive, quantitative, and evaluative. The experimental approach is used to test the performance of the identification model through controlled experimental techniques, while the descriptive approach serves to describe image features and model evaluation results. The quantitative approach is applied to analyze model performance based on predetermined metrics. In addition, an evaluative approach is needed to assess model performance against predetermined research objectives, as well as to draw relevant conclusions. By using this approach, this study aims to provide a comprehensive understanding of model performance in identifying diseases in long bean leaves by utilizing the Convolutional Neural Network (CNN) method.

Dataset

The dataset used in this study was obtained from two different dataset provider websites, namely huggingface.co and Data.mendeley.com. In the initial data collection, this dataset consisted of a total of 1,717 images of bean plant leaves categorized into 6 different conditions. The six categories include leaves in healthy condition (Bean Healthy) as many as 386 images, leaves infected with rust (Bean Rust) as many as 393 images, leaves with spots (Bean Angular Leaf Spot) as many as 389 images, leaves affected by yellow mosaic (yellow mosaic) as many as 220 images, powdery mold as many as 180 images and leaf curl (leaf Crinckle) as many as 150 images. then augmentation was carried out to increase the number of datasets. many images after augmentation were 2316 images divided into 6 classes or 6 labels, in each dataset consisting of 386 images for each class. Bean Healthy 386 images, Bean Rust 386 images, Bean Angular Leaf Spot 386 images, yellow mosaic 386 images, powdery mold 386 images and leaf Crinckle 386 images.

The samples in the dataset are shown in Figure 1 below:

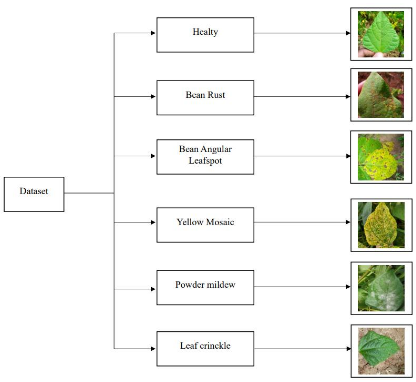


Figure 1. Sample Dataset

The purpose of collecting this dataset is to support further analysis and research on diseases in long bean plants. With the variation in leaf conditions represented by these images, it is hoped that this study can provide deeper insights into diseases that can affect bean plants and effective control strategies.

Data Preprocessing

Data preprocessing is an important stage to process raw data obtained from data sources into high-quality information or have meaningful value (Kamdan et al., 2022). The non-uniformity of data formats is also one of the reasons for conducting data preprocessing, so that data that has been preprocessed can be used to produce accurate data grouping. The data preprocessing stage is carried out to process previously obtained raw data into data that is ready to be used for the model implementation stage. The purpose of this process is to optimize the data to suit the needs of the model and overcome various problems such as lack of data, outliers, and scale variations. In this study, data preprocessing was carried out by setting the batch size, labeling the dataset, performing data splitting and applying image Augmentation.

Design and Selection of Convolutional Neural Network Models

At this stage, the Convolutional Neural Network (CNN) architecture is designed using the ResNet-50 model as a basis. ResNet-50 was chosen because of its ability to handle complex data through the use of shortcut connections, which helps overcome the vanishing gradient problem. Vanishing gradient itself is a problem that occurs when we train a neural network that has many layers. This problem makes the learning process very slow or even stops. This can hinder the network's ability to learn data. So to overcome this, the Resnet-50 model is used to overcome this problem.

Model Training

After the CNN model is designed, the next step is to train the model using the previously collected training data. The model testing process is carried out by setting the number of epochs, which is 100 epochs, to determine how many times the network will undergo the training process. During this stage, the loss function is used to evaluate the performance of the CNN model.

Batch size is a hyperparameter that determines the number of samples to be processed in one iteration during training. Research shows that with a larger batch size, the system can learn more features, thereby increasing the accuracy of the model (Hidayat, 2024). Batch Size refers to the number of groups of sample data used to train the model. In this process, the model will process these samples to remember and learn the data through the iterations carried out (Rizki et al., 2023).

The learning process in artificial neural networks is known as the learning rate. The goal is to optimize the learning parameters to achieve better results (Astria et al., 2022). The learning rate determines how many steps are taken to update the weights. If the learning rate value is too small, the training process will take a long time. Conversely, if the learning rate value is too large, training can be less effective (Rismiyati & Luthfiarta, 2021).

Model Testing Methods

After the model is trained, the next step is to test the model using data that has never been seen before (test data). This test aims to evaluate the performance of the model in identifying diseases in long bean leaves. The evaluation is carried out using a confusion matrix, which helps measure the level of accuracy, precision, recall, and F1-score. Confusion matrix tables are commonly used for binary classification by categorizing the number of correct and incorrect test data (Normawati & Prayogi, 2021). This matrix is arranged in a table, where the actual and predicted classes are represented by rows and columns (Fluorida Fibrianda & Bhawiyuga, 2018). The number of samples predicted to a particular class is indicated by each cell in the matrix, with the main diagonal indicating the number of correctly predicted samples.

System Testing Methods

The system testing in this study uses the black box testing method to ensure that the mobile application developed to detect and classify diseases in long bean leaves functions properly from the user's perspective. Black Box Testing is a stage used to test the smoothness of the program that has been developed. This testing is very important to prevent errors in the program flow (Achmad & Yulfitri, 2020). Black box testing focuses on verifying that the system functions properly from the user's perspective. This method does not evaluate the internal processes of the system, but only assesses the results that can be seen by the user [50]. This testing focuses on the main functionality, such as the user's ability to upload leaf images, image analysis performed by the system, and the presentation of informative results. Test cases are designed to test various conditions, including uploading images of healthy leaves, infected leaves, and irrelevant or low-quality images. The results of this testing are expected to assess the accuracy and performance of the system.

RESULTS AND DISCUSSION

Preprocessing

In this study, the convolutional neural network (CNN) model was applied to identify six classes of leaf conditions of long bean plants, both diseased and unaffected. At this stage, labeling of the dataset uses inferred labels which will provide labels according to the folder names that already exist in the dataset.

```
Found 2315 files belonging to 6 classes.
```

Figure 2 Total of all images in the Dataset

```
['Angular Leaf Spot',
 'Bean Rust',
 'Healthy',
 'Leaf Crinckle',
 'Powdery Mildew',
 'Yellow Mosaic']
```

Figure 3 Labeling On Dataset

Then this dataset is divided into training data, validation data and testing data. With a comparison or percentage of 80% for training data, 15% for validation data and 5% for testing data. Thus the training data consists of 116 batches of training data, 21 batches of validation data and 8 batches of testing data. Where each batch or group consists of 16 images.

```
Length of Training Dataset is 116

Length of Validation Dataset is 21

Length of Testing Dataset is 8
```

Figure 4 Splitting Data

After dividing or splitting the dataset, image augmentation is performed to increase data diversity so that the model has more ability in learning images. Image augmentation is performed by applying several commands to image augmentation, namely RandomFlip, RandomRotation, RandomZoom, RandomTranslation, RandomHeight, and RandomContrast.

Model Selection

At this stage, the selected cnn model is the resnet-50 model. The resnet-50 model itself is a pre-trained model, namely a model that has been created and trained with a large data set called the imagenet dataset so that the model already has an understanding of learning data. We use this model to train using more detailed data so that it can perform detailed identification tasks and according to what we want. In this resnet 50 model, it consists of 50 layers where the layers of all of them except the last 15 layers are given a false value or we do not train them with the dataset we have but use the weights from imaginet. So when the model passes through layers 1-35, the model does not recognize the dataset we use, the model only recognizes the imaginet dataset and in layers 35-50 the model will be trained with the dataset we have so that the model in this layer will better understand and be able to detect plant diseases according to the dataset we have. In addition, this is also done to reduce the considerable computing power because it has to train the 50 existing layers. While actually these 50 layers already have their own intelligence but only need to be adjusted to the new dataset so that they have specific intelligence to carry out identification according to what we expect.

Model Training

At this stage, the dataset is trained using 100 epochs and shows a fairly good graph between the training and validation accuracy graph and the training and validation loss graph as seen in Figure 5 below.

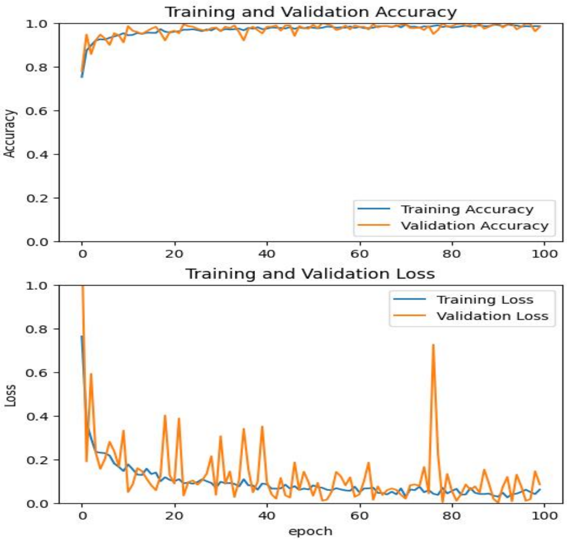


Figure 5. Training dataset results

During training, the training accuracy increased quite well where in the first epoch it had a percentage of 62.7% and continued to increase until it got a final score of 99.11%. while for the accuracy of the validation it had a percentage of 77.95% in the first epoch and got a final score of 98.21% as seen in Figures 6 and 7 below.

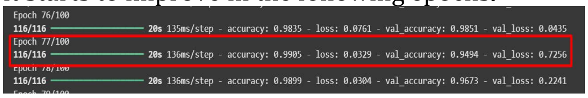
```
Epoch 1/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 50s 272ms/step - accuracy: 0.6277 - loss: 1.0978 - val_accuracy: 0.7795 - val_loss: 1.2022
Epoch 2/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 21s 149ms/step - accuracy: 0.8695 - loss: 0.3935 - val_accuracy: 0.9464 - val_loss: 0.1899
Epoch 3/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 21s 145ms/step - accuracy: 0.8977 - loss: 0.3029 - val_accuracy: 0.8571 - val_loss: 0.5918
Epoch 4/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 22s 152ms/step - accuracy: 0.9297 - loss: 0.2208 - val_accuracy: 0.9196 - val_loss: 0.2312
Epoch 5/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 20s 137ms/step - accuracy: 0.9233 - loss: 0.2195 - val_accuracy: 0.9464 - val_loss: 0.1558
```

Figure 6. Training score on the first epoch

```
Epoch 95/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 26s 178ms/step - accuracy: 0.9899 - loss: 0.0337 - val_accuracy: 0.9702 - val_loss: 0.1301
Epoch 96/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 34s 135ms/step - accuracy: 0.9838 - loss: 0.0506 - val_accuracy: 0.9821 - val_loss: 0.0764
Epoch 97/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 23s 147ms/step - accuracy: 0.9814 - loss: 0.0691 - val_accuracy: 0.9970 - val_loss: 0.0097
Epoch 98/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 25s 182ms/step - accuracy: 0.9807 - loss: 0.0543 - val_accuracy: 0.9940 - val_loss: 0.0177
Epoch 99/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 19s 136ms/step - accuracy: 0.9856 - loss: 0.0399 - val_accuracy: 0.9613 - val_loss: 0.1453
Epoch 100/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 22s 139ms/step - accuracy: 0.9911 - loss: 0.0342 - val_accuracy: 0.9821 - val_loss: 0.0851
```

Figure 7. Training Score On Last Epoch

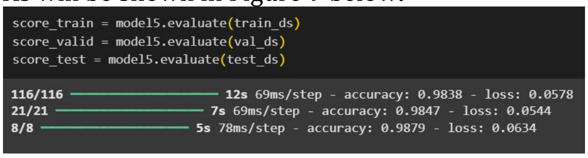
then for the training accuracy and validation loss it looks quite good and gets a decrease from each epoch even though the accuracy of the validation loss in epoch 77 there is an increase of around 72% as seen in Figure 8 but it starts to improve in the following epochs.



```
Epoch 76/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 20s 135ms/step - accuracy: 0.9835 - loss: 0.0761 - val_accuracy: 0.9851 - val_loss: 0.0435
Epoch 77/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 20s 136ms/step - accuracy: 0.9905 - loss: 0.0329 - val_accuracy: 0.9494 - val_loss: 0.7256
Epoch 78/100
116/116 ━━━━━━━━━━━━━━━━━━━━ 20s 136ms/step - accuracy: 0.9899 - loss: 0.0304 - val_accuracy: 0.9673 - val_loss: 0.2241
```

Figure 8. Increase in Validation Loss During Training

Then the training results are evaluated to determine the accuracy of each data. The training results show quite good accuracy, namely 98.3% for training data accuracy, 98.4% for validation data accuracy and 98.7% for testing data accuracy. As will be shown in Figure 9 below.



```
score_train = model5.evaluate(train_ds)
score_valid = model5.evaluate(val_ds)
score_test = model5.evaluate(test_ds)

116/116 ━━━━━━━━━━━━━━━━━━━━ 12s 69ms/step - accuracy: 0.9838 - loss: 0.0578
21/21 ━━━━━━━━━━━━━━━━━━━━ 7s 69ms/step - accuracy: 0.9847 - loss: 0.0544
8/8 ━━━━━━━━━━━━━━━━━━━━ 5s 78ms/step - accuracy: 0.9879 - loss: 0.0634
```

Figure 9. Data Evaluation Results

Model Testing

After passing the training stage, testing is carried out using confusion matrix, precision, recall and f1 score to test how well the model that has been created is in identifying or recognizing diseases in long bean leaves.

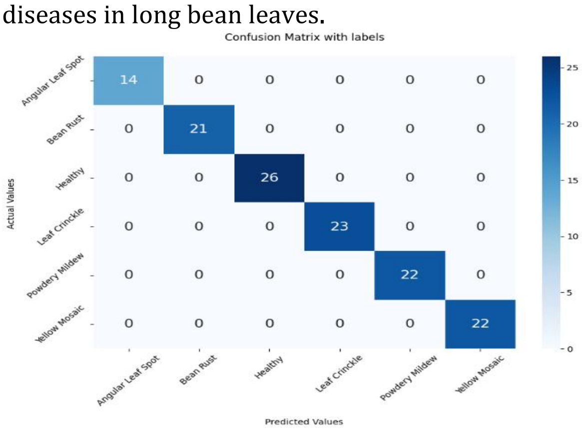


Figure 10. Confusion Matrix test results

In Figure 10 we can see the model testing gets a perfect score where there are no errors in the image identification process. All images are identified according to the labels that have been given. We can see from the 14 bean angular leaf spot data there are no label detection errors in the

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v7i3.373
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

image as well as other disease data all identified according to their respective labels and none are detected in the wrong label. Likewise, the results of the precision, recal and f1-score tests get a perfect score, namely getting a percentage of 100% for each label identification in the image as shown in Figure 11. Thus it can be concluded that the model is good enough in identifying diseases based on images of long bean leaves and the model can be implemented into the application we created.


Classification Report:

	precision	recall	f1-score	support
Angular Leaf Spot	1.0000	1.0000	1.0000	14
Bean Rust	1.0000	1.0000	1.0000	21
Healthy	1.0000	1.0000	1.0000	26
Leaf Crinckle	1.0000	1.0000	1.0000	23
Powdery Mildew	1.0000	1.0000	1.0000	22
Yellow Mosaic	1.0000	1.0000	1.0000	22
accuracy			1.0000	128
macro avg	1.0000	1.0000	1.0000	128
weighted avg	1.0000	1.0000	1.0000	128

Figure 11. Clasification Report

Mobile Based System Implementation

This stage is the stage of implementing the model into a mobile-based system created using the Kotlin programming language with Android Studio as the code editor. The previously created model is a classification model to detect diseases that attack long bean leaves. After the training process is complete, the model is converted into TensorFlow Lite (.tflite) format so that it can be run efficiently on mobile devices. The TFLite format was chosen because it is lighter and compatible with the Android operating system. The model in .tflite format is then integrated into a mobile application consisting of several main views, namely a splash screen as the opening page, a home page as the initial navigation, a prediction page where users upload leaf images, a prediction results page that displays the classification of the model, and an information page that contains explanations of various diseases that commonly attack long bean plants. This model integration allows the application to provide disease predictions directly from images uploaded by users, so that it can help farmers or users in identifying and handling plant diseases more quickly and accurately.

1. **SplashScreen Page**

The SplashScreen page is a page that appears for a few seconds when the application is first run. This page contains the application logo and a welcome message to the application named Dekapan (Long Bean Detection) that has been created. The SplashScreen page will be displayed in Figure 12 below.


Figure 12. SplashScreen Page

2. **Home Page**

The home page is the main page in this application where this page provides detailed information related to long bean plants. Starting from the origin of this plant, then the use of this plant, the potential in the economic sector, to the benefits and nutritional value contained in this plant. The home page will be in Figure 13 below.

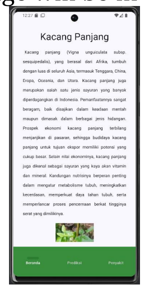


Figure 13. Home Page

3. **Prediction Page**

On this prediction page contains options for predicting diseases on long bean leaves. The options provided in this application include two different ways, namely through the camera on the cellphone used and through images that have been stored in the user's cellphone gallery. The appearance on the prediction page will be shown in Figure 14 below.

Figure 14. Prediction Page

After the user has selected the option and selected the image to be identified for the disease. The user will be presented with a prediction results page where this page contains information on the disease prediction results, prediction scores in the form of percentages and solutions to overcome the disease. The prediction results page will be displayed in Figure 15 below.

Figure 15. Prediction Results Page

4. Disease page

On this disease page, there is information about diseases that usually attack the leaves of long bean plants and the causes of the disease. There are 6 leaf conditions that can be seen on this page, namely leaf conditions in the bean angular leaf spot, bean rust, bean healthy, leaf crincle, powdery mildew and yellow mosaic conditions which are equipped with sample images and information about the causes. The disease menu page will be displayed in Figure 16 below.

Figure 16. Desease Page

System Testing

1) Black Box Testing

This system testing uses the black box testing method to evaluate the performance of the application that has been created. Black box testing is a software testing method that focuses on the function and output of the system without requiring knowledge of the internal structure or source code of the application. This testing is done by providing input and observing the output produced to ensure that the system can function as expected. The following are some of the results of black box testing that have been carried out.

Table 1. System testing using Black Box Testing

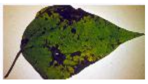
Test Cases	Aspect Assessment	Expected results	Testin Status
Navigation between pages	Click the menu to move from one page to another	Pages can change according to the selected menu.	Passec
Make predictions using the open camera option in the prediction menu	Click the camera option on the prediction menu to ensure the camera option function can be used properly.	The camera option can work well and the camera on the phone can be used to make predictions.	Passed
Make a prediction using the “select from gallery” option in the prediction menu.	Click the select from gallery option on the prediction menu to ensure that the “select from gallery” option function can be used to upload images.	The “select gallery” option can be used to take an image from the user’s phone gallery to make a prediction.	Passed
Upload valid image	This is done by making predictions with several images of diseases on different leaves of long bean plants.	The application can respond well and provide scores for predictions and provide appropriate solutions.	Passed
Ensure that the buttons on the disease page display the appropriat e disease.	This is done by clicking one by one on the buttons listed in the disease types on the disease page.	The application can present sample images and relevant disease information.	Passed

Table 1 black box testing shows that the application of disease identification on long bean leaves has been tested in various scenarios, including navigation testing, camera options for

JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 7, No. 3. Juni 2025 DOI: https://doi.org/10.34288/jri.v7i3.373
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

prediction, select from gallery options for prediction, upload valid images with different disease images, and ensure the button in the disease menu displays the appropriate results. The results show that the application is successful in all test cases, with the status "Passed" for all functionality test scenarios. Then after the system functionality test was carried out, testing was also carried out using new images found on the internet to find out how accurate the system that had been created was in identifying new images that were not in the trained dataset. The results of the identification process that had been carried out will be displayed in Table 2 below :

Table 2. Identification Test Results

New Image		Result	
Test Cases	Aspect Assessment	Expected results	Testin Status
	Bean Rust		Passec
	Bean Angular Leaf Spot		Failed
	Bean Healty		Passec
	Yellow Mosaic		Passec
	Powder Mildew		Passec
	Leaf Crinckle		Passed

The identification results in Table 4.2 show that out of 6 new image data that were identified, there was one error in the identification, namely the angular leaf spots of the bean leaves. The identification results show a confidence score below 70% and indicate an error. The confidence score below 70% is the threshold set to avoid errors in prediction. Although the prediction results may be correct, a low confidence score indicates that the model is not confident enough with its prediction results, which can result in identification errors or failure to recognize the image accurately. However, the system is able to detect new images well and the system can be used.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This study successfully implemented a Convolutional Neural Network (CNN) model to identify six classes of diseases in long bean leaves, using the Resnet-50 Architecture that had been pre-trained on the ImageNet dataset. The dataset consists of 2316 images with each class represented by 386 images, divided into training data (80%), validation (15%) and testing (5%). The ResNet-50 model consists of 50 layers, where the first 35 layers are not trained with the dataset used, but use weights from ImageNet. This means that the model only recognizes the ImageNet dataset in the first 35 layers and in the last 15 layers, the model is trained with a specific dataset to identify diseases in long bean leaves, so that it can better understand and detect diseases according to the data available. This approach also aims to reduce the required computing power, since training all 50 layers at once requires large resources. Model training was carried out by performing 100 epochs. The training results showed quite good accuracy, namely 98.3% for Training data accuracy, 98.4% for validation data accuracy and 98.7% for test data accuracy.

Model evaluation using confusion matrix and reports showed very good performance and no prediction errors. In addition, the trained model also showed good ability in recognizing disease classes in long bean leaves with high precision, recall and f-1 score values so that the model was ready to be implemented into the system created. Then this model was implemented into a mobile-based application to make it easier for users to identify diseases in long bean leaves.

Suggestion

Based on the research that has been done, there are several suggestions that can be applied for further research, including the following:

1. Although this resnet-50 model is quite good at identifying diseases in long bean leaves, it can explore other CNN architectures, such as DenseNet, EfficientNet, MobilenetV2, Inception or other latest CNN models. so that it can produce a model with better accuracy in identifying diseases in long bean plants.
2. This Long Bean Identification Application requires further development by building a more attractive system. In addition, this system can only identify diseases in long bean leaves based on spots, the system can be further developed to be able to detect plant diseases based on shape in order to distinguish leaf types based on shape or detect plant diseases based on other disease indicators such as diseases that attack plant stems or roots, thus expanding the application's ability to identify diseases in plants.

REFERENCES

Achmad, Y. F., & Yulfitri, A. (2020). Pengujian Sistem Pendukung Keputusan Menggunakan Black Box Testisng Studi Kasus E-Wisudawan Di Institus Sains Dan Teknologi Al-Kamal. *Jurnal Ilmu Komputer*, *5*, 42.

Afandi, A. R., & Kurnia, H. (2023). Revolusi Teknologi: Masa Depan Kecerdasan Buatan (AI) dan Dampaknya Terhadap Masyarakat. *Academy of Social Science and Global Citizenship Journal*, *3*(1), 9–13. https://doi.org/10.47200/aossagcj.v3i1.1837

Arafa, N. P., Basri, S. R., & Saputra, R. A. (2024). *KLASIFIKASI PENYAKIT PADA DAUN TANAMAN CABAI DENGAN PENDEKATAN ARTIFICIAL NEURAL NETWORK (ANN)*. *8*(6), 12865–12871.

Astria, C., Windarto, A. P., & Damanik, I. S. (2022). Pemilihan Model Arsitektur Terbaik Dengan Mengoptimasi Learning Rate Pada Neural Network Backpropagation. *JURIKOM (Jurnal Riset Komputer)*, *9*(1), 109. https://doi.org/10.30865/jurikom.v9i1.3834

Azka, N. A., & Sayekti, R. S. (2020). KARAKTERISASI AKSESI KACANG PANJANG (Vigna unguiculata subsp. sesquipedalis) LOKAL. *Agrotechnology Innovation (Agrinova)*, *3*(2), 14. https://doi.org/10.22146/a.62709

Badan Pusat Statistik indonesia. (2024). *Produksi Tanaman Sayuran, 2021-2023*. Bps.Go.Id. https://www.bps.go.id/id/statistics-table/2/NjEjMg==/production-of-vegetables.html

Fluorida Fibrianda, M., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(9), 3112–3123. http://j-ptiik.ub.ac.id

Hidayat, Z. S. (2024). *PENENTUAN UKURAN BATCH OPTIMAL UNTUK PELATIHAN YOLOV8 DALAM PENDETEKSIAN OBJEK PADA KENDARAAN OTONOM DETERMINATION OF OPTIMAL BATCH SIZE FOR YOLOV8*. *9*(1).

Hidayati, N., Husna Nurrohmah, S., & Ardhany, F. (2020). Isolasi, identifikasi dan karakterisasi penyebab penyakit bercak daun pada semai pinus di perum perhutani BKPH Purworejo, KPH Kedu Selatan. *Jurnal Pemuliaan Tanaman Hutan*, *14*(1), 21–32. https://doi.org/10.20886/jpth.2020.14.1.21-32

Kamdan, Ivana Lucia Kharisma, Gina Purnama Insany, & Paikun. (2022). Research topic modeling in informatics engineering study program at Nusa Putra University using LDA method. *International Journal Engineering and Applied Technology (Ijeat)*, *5*(2), 24–35. https://doi.org/10.52005/ijeat.v5i2.76

Kusuma, B. M., Hermanto, T. I., & Lestari, D. (2025). *Klasifikasi Jenis Penyakit Pada Tanaman Padi Menggunakan Algoritma Convolutional Neural Network Classification of Paddy Rice Plant Diseases Using Convolutional Neural Network Algorithms*. *1*, 40–52. https://doi.org/10.26798/jiko.v9i1.1395

Lesmana, A. M., Fadhillah, R. P., & Rozikin, C. (2022). Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN). *Jurnal Sains Dan Informatika*, *8*(1), 21–30. https://doi.org/10.34128/jsi.v8i1.377

Maysela, A., & Rohma, N. (2024). Diagnosa Penyakit Tanaman Tomat pada Citra Daun Menggunakan Metode Convolutional Neural Network (CNN). *JIMU: Jurnal Ilmiah Multi Disiplin*, *02*(03), 555–567. https://ojs.smkmerahputih.com/index.php/jimu/article/view/407

Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen

Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(2), 697–711.

Peryanto, A., Yudhana, A., & Umar, R. (2020). Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing*, *4*(1), 45–51. https://doi.org/10.30871/jaic.v4i1.2017

Putra, I. P., Rusbandi, R., & Alamsyah, D. (2022). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network. *Jurnal Algoritme*, *2*(2), 102–112. https://doi.org/10.35957/algoritme.v2i2.2360

Putra, J. V. P., Ayu, F., & Julianto, B. (2023). Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN. *Stains (Seminar Nasional Teknologi & Sains)*, *2*(1), 155–162. https://proceeding.unpkediri.ac.id/index.php/stains/article/view/2888

Rismiyati, R., & Luthfiarta, A. (2021). VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification. *Telematika*, *18*(1), 37. https://doi.org/10.31315/telematika.v18i1.4025

Rizki, F., Kharisma Putra, M. P., Assuja, M. A., & Ariany, F. (2023). Implementasi Deep Leraning Lenet Dengan Augmentasi Data Pada Identifikasi Anggrek. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, *4*(3), 357–366. https://doi.org/10.33365/jatika.v4i3.3652

Setyadi, R. A., Rahman, S., Manurung, D., Hasanah, M., & Indrawati, A. (2024). Implementasi Transfer Learning Untuk Klasifikasi Penyakit Pada Daun Cabai Menggunakan Cnn. *Djtechno: Jurnal Teknologi Informasi*, *5*(2), 304–315. https://doi.org/10.46576/djtechno.v5i2.4642

Suprihanto, S., Awaludin, I., Fadhil, M., & Zulfikor, M. A. Z. (2022). Analisis Kinerja ResNet-50 dalam Klasifikasi Penyakit pada Daun Kopi Robusta. *Jurnal Informatika*, *9*(2), 116–122. https://doi.org/10.31294/inf.v9i1.13049