International Journal of Informatics and Computation (IJICOM)
Vol.6, No.2, December 2024
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Mobile Application Development for Chili Disease Detection with Convolutional Neural Network

Sri Winiarti[1], Itsnaini Irvina Khoirunnisa [2], Norhudah Seman[3]

Abstract

The demand for chili continues to increase along with population growth and the industrial sector, but the supply is unstable due to weather factors such as high rainfall and humidity. This condition causes the spread of diseases in chili plants such as anthracnose fruit rot, begomovirus yellow virus, and leaf spots. This study aims to develop a chili plant disease identification system and evaluate the accuracy of chili plant disease image classification. This study is expected to help experts provide recommendations for accurately controlling the spread of disease, assisting farmers in identifying diseases early, and improving the quality and quantity of chili plant harvests. The method used is Transfer Learning using the Convolutional Neural Network (CNN) MobileNet V2 architecture. The research stages include data collection, needs analysis, preprocessing design, model design, architecture stages, accuracy testing, system implementation, and system testing. These stages are carried out sequentially without any being skipped. Accuracy testing is calculated using a confusion matrix. In addition, the system that has been created will be tested by functional and user testing. Chili plant disease image data was obtained from chili plantations in Sumowono District, Semarang Regency. A dataset of 4,500 chili plant disease images was used, and 70% of the data was divided into training data and 30% for validation data. The accuracy results obtained were 99% in the training process and 94% in the validation process. Model evaluation using a new dataset of 150 chili plant disease images showed an accuracy result of 94%. Functional testing and user testing by 10 farmers produced an average value of 90. Thus, it can be concluded that the system can identify chili plant diseases well, as well as support agricultural activities and farmer needs.

Keywords:
Chili Plant Disease, Convolutional Neural Network, System Mobile.

1. Introduction

Please note that the first paragraph of a section or subsection is not indented. The first paragraphs following a table, figure, equation, etc. do not have an indent. Chili (Capsicum sp.) is a plant used in food, spices, traditional medicine, and others. Chili peppers include fruits and vegetables with high protein content and favorable economic value [1]. Following the growth of the population and industrial sectors' development, the demand for chili peppers for food needs is increasing [2]. Unfortunately, the demand for chili does not match the stable supply of chili, causing fluctuations. Planting chili plants in the rainy season will face weather obstacles that are unsafe for plant growth. Rainfall and high air humidity cause the development of chili plant diseases such as anthracnose fruit rot (Colletotrichum sp.), yellow virus (Begomovirus), and leaf spot [3].

Corresponding Author: Sri Winiarti (sri.winiarti@tif.uad.ac.id)
1 Itsnaini Irvina Khoirunnisa, Universitas Ahmad Dahlan. itsnaini2000018274@webmail.uad.ac.id
2 Norhudah Seman, Universitas Ahmad Dahlan.

In CNN, there is a Transfer Learning method. Transfer Learning is a term that refers to the use of knowledge from one field to another so that pre-trained models can learn new things [4]. By using Transfer Learning, you can retrain models that have been trained previously with the desired dataset. One of the architectures contained in Transfer Learning is MobileNet V2. This architecture can produce high accuracy; besides, it is lightweight and can be run on computers with not too high performance [5].

This study identifies chili plant diseases based on leaves and fruits because identification can provide results based on the available data, namely the dataset of types of chili plants collected. In this study, chili plant diseases were identified based on leaves and fruits and did not consider other factors such as roots and stems, because each chili plant variety has different characteristics [6]. Object classification is carried out to distinguish image characteristics that can be used as an aid in diagnosing chili plant diseases by studying patterns from plant disease research data. The data is used to create a model that can distinguish whether the plant is infected with a disease or not. So that it can be classified based on the proximity of old and new information [7]. The reason this study uses the Transfer Learning method with the MobileNet V2 Architecture is that it can be used to perform classification with high accuracy. MobileNetV2 is an architectural model that can be used on mobile devices. This is the background to the selection of the MobileNetV2 architecture in this study which aims to create an application for detecting chili plant diseases.

2. Related Works

Research [8] states that disease attacks on chili plants can damage plants and cause crop failure. Farmer lack of knowledge of farmers about diseases that attack chili plants causes delays in the diagnosis and treatment of diseases. Farmers are still constrained by the lack of recommendations for effective control methods, so they tend to use chemical pesticides that hurt the environment [9].

Research [1] presents the use of Deep Learning to detect chili plant diseases using leaf images from chili plants taken directly by researchers. This research was conducted because many chili plants were attacked by diseases due to unpredictable climate change, causing crop failure. This research used the Rasberry Pi Camera to build a detection system on chili plants. For the detection system to work well, there are three limitations: light, distance, and object. After testing, it was found that with sufficient light and a distance of less than one meter, high accuracy reached 100%, but with lower light and a distance of more than one meter, the accuracy was 68.8%.

Research [9] presents using the Convolutional Neural Network (CNN) method to detect diseases in chili plant leaves. The results make it possible to classify images of chili plant leaves taken directly from the garden. This research was conducted because the disease can potentially develop, resulting in plant damage. This research uses data collected directly from chili plantations in Central Java. This research can automatically classify four leaf conditions: yellow virus disease, mosaic curl disease, wilted leaves, and healthy leaves. The accuracy result obtained using AlexNet architecture is 90%.

Research [10] uses CNN methods to classify melon leaf diseases through plant labeling. Some farmers do not realize the attack of melon leaf disease. Flies, ticks, and other organisms cause diseases that attack melon leaves. In addition, this research also uses Computer Vision to analyze the input image. The dataset is taken directly from melon plantations in Sugio District, Lamongan Regency, and contains as much as 10,000 data. Data collection is carried out based on the age of melon plants. The time range has been determined for testing with a computer system with 89% accuracy and an Android smartphone system with 90% accuracy.

Research [11] presents the Deep Learning CNN method by utilizing AdaGrad, Adam, RMSProp, Adadelta, Adamax, and SGD algorithms to classify images of corn leaf disease. This research uses much image data to sort diseases more simply, quickly, and efficiently. In this research, researchers obtained datasets from rice fields and took pictures of 900 image data with a cellphone. Testing six architectures resulted in varying accuracy. Adam's algorithm produces the highest accuracy of 84%, Adadelta produces the lowest accuracy of 65%, Adamax produces 83% accuracy, RMSProp produces 82% accuracy, SGD produces 78% accuracy, and AdaGrad produces 75% accuracy. Based on the accuracy of the six algorithms, classification using Adam's algorithm is the best method for detecting diseases on corn leaves.

3. Proposed Method

The flow chart for this research includes several main stages, starting with data collection, preprocessing, model design, architecture stages, results, system implementation, and system testing. Figure 1 shows details of the flow chart describing the overall flow of this research.

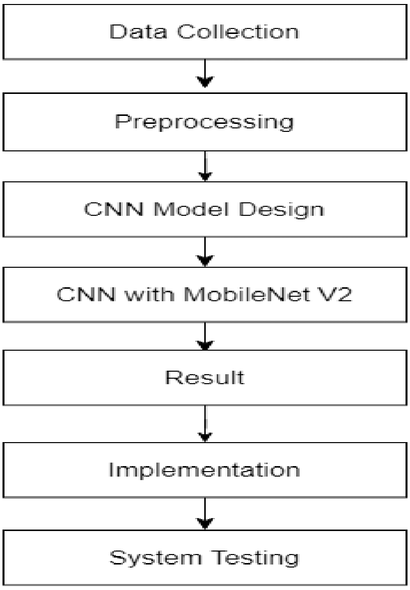

Fig. 1. Research Stages

3.1. Image Dataset

The image dataset of chili plant diseases was obtained directly from chili plantations in Sumowono District, Semarang Regency. The image dataset consists of 3 disease classes: anthracnose fruit rot, begomovirus fruit rot, and leaf spot. The total dataset has 4,500 images, with each disease class having 1,500 image data. The image dataset was taken at close range, which allows details of disease symptoms on chili plants, and at long range, which can capture the overall condition of chili plants in an environment. Figure 2 shows an example of the original image, which is 3,000 × 3,000 pixels in size.

a) Fruit Rot Due to Anthracnose | b) Begomovirus Yellow Virus | c) Leaf Spot

Fig. 2. Chili Plant Disease Data Collection

The image dataset of chili plant diseases used in this research is divided into two parts, with a proportion of 70% testing data and 30% validation data. In addition, 150 new image datasets were used as testing data. Details of the distribution of the chili plant disease dataset can be seen in Table 1.

Table 1. Dataset Distribution

Type of Disease	Training Data	Validation Data	Testing Data	Total Data
Fruit Rot Due to Anthracnose	1350	150	50	1550
Begomovirus Yellow Virus	1350	150	50	1550
Leaf Spot	1350	150	50	1550

3.2. Image Preprocessing

Before being used to train the CNN model, training and validation data undergo a preprocessing stage in the form of resizing. This resize process is done to equalize the size of all images in the dataset, initially 3,000 × 3,000 pixels to 256 × 256 pixels. The purpose of this process is to make it easier for the CNN model to find consistent visual patterns in all data without being constrained by differences in image size [12]. The CNN model has difficulty consistently identifying relevant features if the dataset has various sizes. Therefore, resizing all images to the same size is essential in the preprocessing stage.

3.3. Convolutional Neural Network

CNN is a type of deep learning because of the depth of the network, which is often applied in image data processing [13]. The CNN model used in this research is a standard convolution model consisting of three layers: the convolutional layer, pooling, and fully connected layer [14]. In this research, the CNN model used has several configuration parameters. The activation functions used are ReLU (Rectified Linear Unit) and Softmax. ReLU was chosen because it can increase the non-linearity of the overall network and decision function without affecting the convolutional layer aspect [15]. The ReLU equation is shown in equation 2.1 [16].

$$f(x) = x^{+}$$
$$= \max(0, x) = \frac{x + [x]}{2}$$
$$= \begin{cases} x, & if\ x > 0, \\ 0 & otherwise \end{cases} \quad \ldots\ldots (1)$$

Derivative function is

$$f(x)' =$$
$$= \begin{cases} x, & f\ or\ \ x < 0, \\ 0, & f\ or\ x\ \leq 0 \end{cases} \quad \ldots\ldots (2)$$

Meanwhile, the Softmax activation function is applied to neural networks that have multi-class output categories. This function will replace numerical and probability values with a specific proportional scale. In this way, the class with the highest probability value will be selected, and the data entered into it will be classified as that class [9]. The SoftMax activation has the following form [17]:

$$S(x_1 \ldots x_n) = \frac{\exp\,(x)_i}{\sum_1^n exp(x)_n} \quad \ldots\ldots\ldots (3)$$

Regarding the model training process, the number of epochs chosen is 10, meaning that the model will perform the training and validation process for ten iterations. The CNN model in this research consists of 3 layers, each using a 3 × 3 kernel. The convolution filters in each layer are 32 in the first layer, 64 in the second layer, and 128 in the third layer. The learning rate used in the training process is 0.001. The image classification process using the CNN model is carried out in 2 stages. The first stage is the training and validation process, while the second stage is the testing process. The CNN method used in this research is depicted in the flowchart, as shown in Figure 3.

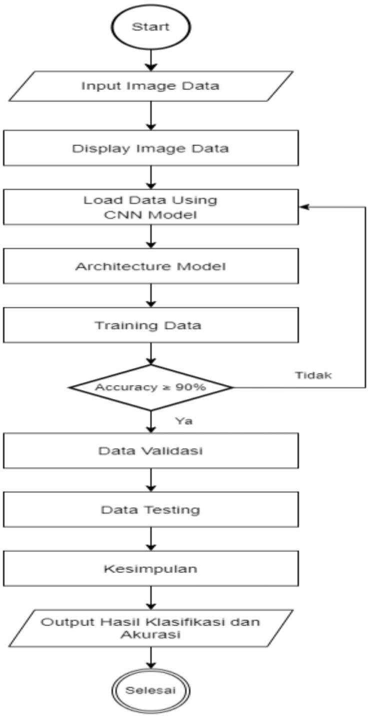

Fig. 3. Flowchart of CNN Process

3.4. MobileNet V2

The next step is to build a CNN architecture to classify chili plant disease images. In this research, the CNN architecture used is MobileNet V2, called through the Keras TensorFlow library. MobileNet V2 is a CNN model with a reasonably high accuracy score, several training parameters, and a miniature model size, but it still performs well [18]. MobileNet V2 has a similar architecture to MobileNet V1 [19]but adds two new CNN layers, namely the inverted residual with a linear bottleneck module [20]. MobileNet V2 is used because this architecture is designed to handle the needs of limited computer resources, such as mobile devices [21]. Using MobileNet V2, researchers can build CNN models that can accurately classify chili plant diseases but can still be run on mobile devices with good performance. A summary of this architecture can be seen in Table 2.

Table 2. Summary of MobileNet V2 Architecture

Descriptions	**Total**
Total Params	2.261.827 (8.63 MB)
Trainable Params	3843 (15.01 KB)
Non-Trainable Params	2.257.984 (8.61 MB)
Input Size (MB)	0.75
Forward/Backward Pass Size (MB)	17.26
Params Size (MB)	8.63
Estimated Total Size (MB)	26.63

3.5 Classification of Model Evaluation

Calculation of accuracy in classification is an important step in the effective use of the method used. Testing data that is not involved in training can be used as evaluation data. This stage shows how well the use of the CNN method can classify chili plant diseases. The evaluation is done using the confusion matrix, which is shown in Table 3. A confusion matrix often known as a table consisting of actual and predicted data from each observed class [22].

Table 3. Confusion Matrix

		Prediction	
		True	**False**
True Data	Positive	TP	FP
	Negative	TN	FN

Precision is a measure of certainty, that is, what percentage of tuples labeled as positive are actual in reality. Equation 2 calculates precision.

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Precision is a measure of certainty, that is, what percentage of tuples labeled as positive are true in reality. Equation 2 calculates precision.

$$presisi = \frac{TP}{TP + FP} \quad (5)$$

Recall is the percentage of tuples that are labeled as positive. The recall is commonly known as the completeness measure. Equation 3 calculates recall.

$$recall = \frac{TP}{TP + FN} \tag{6}$$

F1-Score is the percentage ratio between the harmonic mean of precision and recall [11]. Equation 4 calculates the F1-Score.

$$F1 - Score = \frac{2 \times Presisi \times Recall}{Presisi + Recall} \tag{7}$$

3.6 Implementation

In this research, the application is designed using a mobile system. The CNN model that has been trained will be integrated with TensorFlow Lite, thus allowing implementation in mobile applications [23]. This application is built using the Kotlin programming language. The designed application will display the results of the images entered by the user and predict the type of chili plant disease and its symptoms. Thus, this application can help users identify chili plant diseases on a mobile basis.

3.7 Mobile System Testing

The testing process is carried out to ensure the application runs according to the system design. Application testing is done in two ways: functional testing to ensure each application feature works correctly and user testing using the System Usability Scale (SUS) method to evaluate user usability and satisfaction with the application [24].

4. Experimental Setup

CNN is included in the Deep Learning type because the depth of its network is often applied in image data processing. CNN uses a three-dimensional architecture, namely width (weight), height (height), and depth (depth). As in Figure 2.1, each CNN layer converts the three-dimensional input volume into a three-dimensional output volume of neural activation [21].

The image data used as input consists of height × width × color depth. The image size in pixels is represented by height and width. Meanwhile, the color depth represents the Red, Blue, and Green (RGB) channels. CNN has weight, activation function, and bias like other neural networks. These functions allow the neural network to produce output that matches the training data provided. The softmax activation function is applied to a neural network with a multi-class output category to produce classification results. This function replaces numeric values with probability values with a specific proportional scale. The values generated by this use are considered unnormalized probabilities for each class. In this way, the class with the highest probability value will be selected, and the data entered into it will be classified as that class. If graphed, the softmax function will look like Figure 4.

The softmax activation function is used to obtain the classification results. The activation function produces values that are interpreted as unnormalized probabilities for each class. The class values are calculated using the softmax function, shown by Equation (1) [25].

$$y_{ijk} = \frac{e^{x_{ijk}}}{\sum_{t=1}^{D} e^{x_{ijt}}} \quad \ldots\ldots (8)$$

In this way, the class with the highest probability value will be selected, and the data entered into it will be classified as that class. If graphed, the softmax function will look like Figure 4.

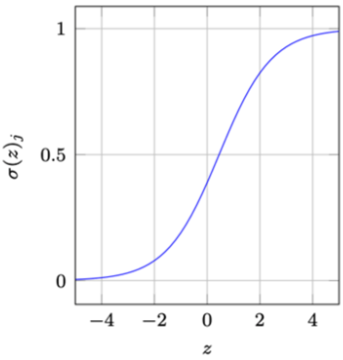

Fig 4. Softmax Activation Function

This study processed data after taking pictures of chili directly using a smartphone. The image will be processed using the methods; resize and augmentation. Resize changes the image size. The previous image size had an initial resolution of 3,000 × 3,000 pixels changed to 256 × 256 pixels. This aims to equalize the size of the image data that has been collected previously. Augmentation seeks to increase the amount of image data for the training process.

The data analysis carried out with the system to be created is a mobile application that can identify images of diseases in chili plants. The system requirements are as follows.

1. Input Requirements
   The data entered into the system is chili plant disease image data with *.jpg format that has sufficient resolution.
2. Process Requirements
   In the process of identifying chili plant diseases, a CNN model that has been trained previously is used.
3. Output Requirements
   The output is the identification of types of chili plant diseases, which are divided into three types: anthracnose fruit rot, Begomovirus Yellow virus, and fruit spots. The output includes the type of disease and symptoms.

5. Result and Analysis

5.1. Convolutional Neural Network

In this study, training and data validation were conducted using image datasets. The training image dataset was 3150 and the validation dataset was 1350 in *.jpg format, divided into three classes. The training dataset in each class consisted of 1050 images, while the testing data for each class consisted of 450 images. At this stage, training was conducted using the CNN model architecture created in Figure 4. The results of training the CNN model architecture that had been created using training and valid data using 10 epochs are shown in Figure 4. The model will iterate 10 times through training and validation data. After ten epochs, the resulting model shows good performance. On the training data, the model has an accuracy of 99% and a loss of 2.7%. This indicates that the model can classify the training data accurately and only has a few errors.

Meanwhile, regarding the validation data, the model achieved 94% accuracy and 28%

loss. Although the accuracy is slightly lower than the training data, this result still performs well. The graph in Figure 4 shows an increase in accuracy towards a value of 1 and a decrease in loss towards a value of 0 during the training and validation process. This indicates that the model can learn the patterns in the data and make predictions with high accuracy on new data that has never been seen before.

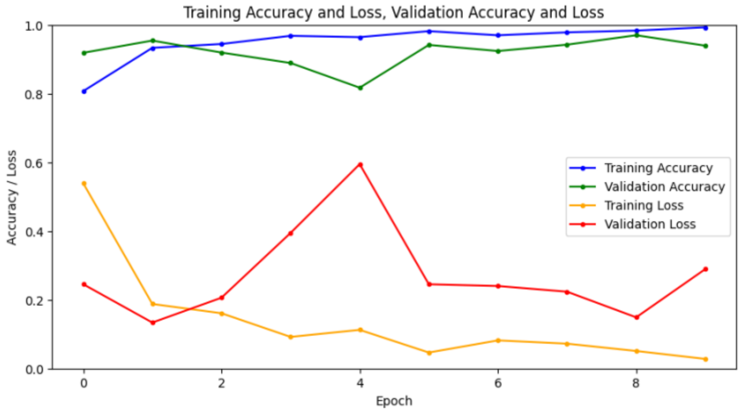

Fig. 4. Training and Validation Charts

An evaluation was conducted using validation data to evaluate the performance of the CNN model in classifying chili plant diseases. The review is done by calculating several metrics: accuracy, precision, recall, and F1-score. The results of the CNN model performance evaluation are shown in Figure 5. These metrics are used to measure how well the CNN model classifies chili plant diseases based on the image dataset used in the research.

Classification Report :

	precision	recall	f1-score	support
antraknosa	0.95	0.93	0.94	450
begomovirus	1.00	0.94	0.96	450
bercakdaun	0.88	0.95	0.91	450
accuracy			0.94	1350
macro avg	0.94	0.94	0.94	1350
weighted avg	0.94	0.94	0.94	1350

Figure 5. Results of the confusion matrix evaluation

At this stage, the researcher determines the best model to obtain optimal results. This is done by finding the best value of the parameters in the CNN model such as input to the image, the effect of kernel size on the convolution layer, and the effect of the number of convolution layers. The purpose of determining the model parameters is to find out the model that has the best results as shown in Table 4.

Table 4. *Trial and Error* Parameter

Parameter Input Shape		
Input Shape	Accuracy Validation	Loss Validation
64 × 64	97%	0,0726
128 x 128	93%	0,2742
256 x 256	94%	0,2982
Kernel Count Parameter		
2 x 2	95%	0,2000
3 x 3	94%	0,2892
5 x 5	93%	0,3089
Parameter Number of Convolution Layers		
2	88%	0,3666
3	94%	0,2892
4	96%	0,2025

Based on Figure 5, it can be seen that the CNN model shows good performance in classifying chili plant diseases. The accuracy, precision, recall, and F1-score values are high, close to 1. This indicates that the CNN model used in this research can accurately identify and classify the types of diseases in chili plants.

Next, the testing stage is carried out to evaluate the performance of the CNN model that has been built. For the testing stage, 150 new image datasets are used, which are divided into 50 images for each class of chili plant disease. The results of testing the CNN model on the new image dataset are shown in Figure 6. Based on the testing results, the CNN model achieved an accuracy of 94%. This high accuracy indicates that the CNN model built has good generalization capabilities and can accurately classify chili plant diseases on new image datasets that have never been seen before. The results of the validation test are shown in Figure 6.

The analysis will focus on the trend of accuracy across different learning rates to identify the optimal rate for achieving the highest accuracy. Based on the data and visualizations provided, I will summarize the findings.

Based on the data and graphs that have been created, here is the analysis related to Accuracy:

1. Trend of Accuracy:
   Accuracy increases significantly when the learning rate decreases from 0.01 to 0.001. At learning rates of 0.001 to 0.00001, accuracy remains high, with the highest value at 0.00001 (94.67%).
2. Optimal Learning Rate:
   A learning rate of 0.00001 gives the highest accuracy (94.67%), indicating that the model performs best at minimal learning rates.
3. Conclusion:
   Smaller learning rates tend to produce higher accuracy, but the difference becomes small after 0.001.

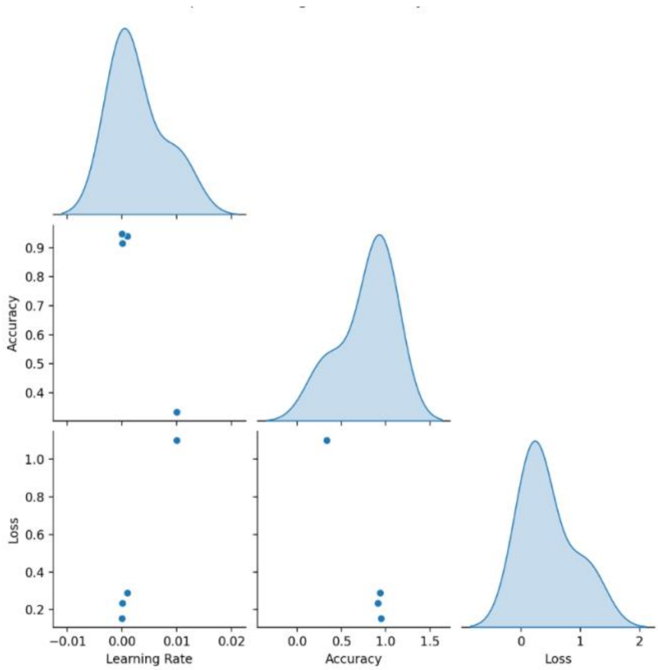

Fig. 6. The Validation, loss, and Accuracy Chart

The pair plot and heatmap provide insights into the relationships and correlations between learning rate, accuracy, and loss. I will now explain their significance for validation analysis. Based on Figure 6, it can be explained that the relationship between Learning Rate, Accuracy, and Loss has resulted in a negative relationship between Learning Rate and Accuracy: the smaller the learning rate, the higher the accuracy. A negative relationship is also seen between Learning Rate and Loss: the smaller the learning rate, the lower the loss. Accuracy and Loss have a strong negative relationship, indicating that as accuracy increases, loss decreases. There is a strong negative correlation between Learning Rate and Accuracy (-0.97), and Learning Rate and Loss (0.99). A robust negative correlation between Accuracy and Loss (-0.99), confirms the inverse relationship between the two. Here is the heatmap visualization and analysis for the validation of input shape, kernel size, and convolution layers in Figure 7.

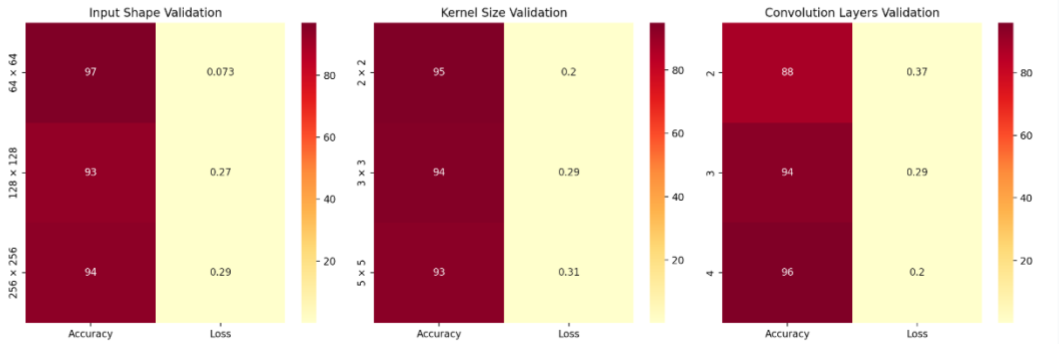

Fig. 7. Testing Results of New Dataset

Validation Analysis as:

1. Input Shape:
   Best performing: 64x64 (Accuracy: 97%, Loss: 0.0726)
   The smaller size gives better results

2. Kernel Size:
   Best performing: 2x2 (Accuracy: 95%, Loss: 0.2000). The smaller kernel size shows better performance
3. Convolution Layers:
   Best performing: 4 layers (Accuracy: 96%, Loss: 0.2025). More layers increase accuracy and decrease loss. This analysis highlights the best-performing parameters for each category and their impact on accuracy and loss

Based on the data analysis that has been done, here are the advantages of CNN in the data:

1. Input Shape Optimization
   64x64 input shape achieves the highest accuracy (97%) with the lowest loss (0.0726). Proving CNN is efficient in processing small-resolution images without losing important information
2. Kernel Size Effectiveness
   2x2 kernel gives the best results (Accuracy 95%, Loss 0.2000). CNN can detect important features with a small kernel, showing efficiency in feature extraction.
3. Layer Architecture
   Performance increases with the addition of layers (4 layers: Accuracy 96%, Loss 0.2025). Demonstrates CNN's ability in complex hierarchical feature learning.
4. Model Stability
   Accuracy consistency above 90% in various configurations. Relatively low loss (<0.4) indicates a stable model.

5.2. Model Implementation on Mobile System

A mobile application system was developed to implement the previously built CNN model. Before integrating the model into the mobile application, the CNN model was first integrated into Tensorflow Lite. By converting the CNN model into Tensorflow Lite, the model can be run efficiently on mobile devices without significant loss of accuracy. The application is designed to accept a new image as input, and the trained model will perform predictions on the image. Although the application developed to detect chili plant diseases cannot perform real-time detection, it is equipped with several features that make it easier for users to classify images of chili plant diseases.
This application has three main features:

1. There is an image select feature that can be used to select the image you want to predict.
2. Users can use the detection feature to make predictions on images that have been inputted into the application.
3. The application provides a display feature that functions to display the image that has been inputted and displays the type of disease detected along with the symptoms.

With these features, users can easily use the application to identify chili plant diseases based on their images. The flowchart of this mobile application is shown in Figure 8, which provides a detailed description of the workflow of the application system.

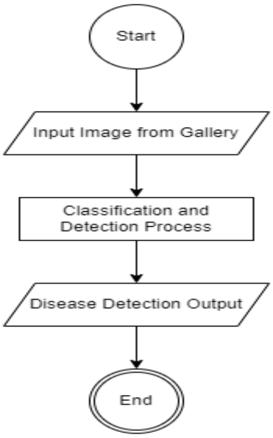

Fig. 8. Flowchart of Mobile System

An overview of the mobile system is shown in Figure 9.a. The mobile application provides various options and functions that the user can explore. One of the main features of this application is the ability to detect diseases in chili plants. Selecting an image allows the user to access the image storage gallery on the smartphone device. Through this feature, users can choose images of chili plants that match the condition of the plants they want to detect.

After the image is selected, the user can make predictions on the image. This mobile application can analyze images inputted by users and provide prediction results related to the condition of chili plants. When the user uploads a new image into the application, the prediction process will be carried out immediately. The prediction process is carried out using a CNN model that has been previously trained with a dataset of chili plant disease images.

The prediction results will be visualized in a display box on the application interface, as shown in Figure 9.b. Through this feature, users can easily find out the condition of the chili plants, whether indicated by disease or not. Visualizing the prediction results in a display box makes it easier for users to understand and interpret the application's findings intuitively. With the ability of digital image-based prediction, this mobile application can help users conduct early identification of chili plant diseases, thus becoming a tool for chili farmers to monitor plant health.

Fig. 9.a Visualization of Mobile Applications

Fig. 9.b Classification Results of Mobile Applications

5.3 System Testing

System testing of mobile applications is done in two main ways. First is functional testing to ensure the developed system can function properly and meet user needs. This testing is done using predefined test case scenarios to test the features and functions in the mobile application. Each test case has specific and detailed testing steps. Each scenario is systematically executed to verify whether the application can perform tasks as expected, such as image upload, image analysis, and prediction result display.

The results show that the application can function properly, such as redirecting to the main page, displaying a message if you have not selected an image, detecting the disease type, and displaying related information based on the selected image. The application can also handle irrelevant images by displaying a message that the image is not a type of chili plant disease. This functional testing is the basis for ensuring the quality and reliability of the application before user testing.

The second test is user testing using the SUS method. This test was conducted by ten farmers, including farmers who can use technology well and who have limitations in operating technology. Farmers were allowed to use the mobile application and give a score on a scale of 1 to 5 for each question contained in the SUS method. These questions cover ease of use, efficiency, and user satisfaction with mobile applications. The data collected from farmers is then calculated using the rules in the SUS method calculation. From the test results, the average score of the results from ten farmers is 90. The score falls into the excellent category with an assessment level of B on the SUS scale. This shows that based on testing with the SUS method, the mobile application system tested gets a good assessment and is suitable for use.

From the two tests that have been carried out, the mobile application system for detecting chili plant diseases based on images has been successfully developed. The application can function according to user needs and has a level of usability that is feasible for farmers to use. With its ability to detect chili plant diseases based on images, this application can assist farmers in identifying diseases early and precisely, thus enabling more effective and efficient treatment. These results prove that the application can significantly support agricultural activities and farmers' needs.

6. Conclusion

Based on the research that has been done on the design and testing of models using the Convolutional Neural Network (CNN) algorithm with the Transfer Learning method using the MobileNet V2 architecture for disease detection in chili plants, the following conclusions were obtained:

1. The level of accuracy in the CNN model using training data of 3150 images and validation data of 1350 images, which are divided into three classes of chili plant diseases. The accuracy results obtained were 99% in the training process and 94% in the validation process. It can be concluded that the application of the CNN method can classify chili plant disease images well. This study uses 150 new images for testing using the model that has been created. The test results obtained an accuracy of 94% in detecting chili plant diseases.
2. based on the analysis of the validation test, the results obtained are that Input Shape Optimization 64x64 input shape achieves the highest accuracy (97%) with the lowest loss (0.0726). Proving CNN is efficient in processing small-resolution images without losing important information and Model Stability with Accuracy consistency above 90% in various configurations. Relatively low loss (<0.4) indicates a stable model.

Acknowledgment

Thanks to the Research Institute of University Teknologi Malaysia as a partner in this research for the funding grant provided to Ahmad Dahlan University, which supports the implementation of this research. Thanks to the Research and Community Service Institute of Ahmad Dahlan University for their support, guidance, and direction until the completion of this research. To all research team members, thank you for your hard work, support, time, thoughts, and energy.

References

[1] R. Rosalina and A. Wijaya, "Detection of Diseases in Chili Leaves Using the Deep Learning Method," *Jurnal Teknik Informatika dan Sistem Informasi,* vol. 6, no. 3, 2020.

[2] L. Marianah, " Vector Insects and Intensity of Virus Diseases in Red Chili Plants," *AgriHumanis: Journal of Agriculture and Human Resource Development Studies,* vol. 1, no. 2, p. 127–134, 2020.

[3] S. Muhamad, Tips for Successful Chili Harvesting Throughout the Season, Jagakarsa, Jakarta: PT AgroMedia Pustaka, 2018.

[4] H. Harmiansyah and et al, " Smart Disease Detection System in Robusta Coffee Plants Using SSD MobileNet V2 as a Pre-Trained Model," *Agrikultura,* vol. 34, no. 1, p. 154–162, 2023.

[5] W. Hastomo and Sudjiran, "Convolution Neural Network Architecture Mobilenet-V2 For Detecting Brain Tumors," in *Seminar Nasional Teknologi Informasi dan Komunikasi STI&K (SeNTIK)*, 2021.

[6] S. Winiarti, M. Saputro and M. S. S, " Deep Learning in Identifying Heritage Building Types with Convolutional Neural Network Algorithm," *Jurnal Media Binadharma,* vol. 5, no. 3, p. 831–837, 2021.

[7] H. D. Abdullah Hamid, N. Hidayat and R. Kartika D, "Diagnosis of Chili Plant Diseases Using the MKNN Method," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* vol. 3, no. 3, p. 2881–2886, 2019.

[8] R. Dzaky, A. Tsany and I. , " Chili Plant Disease Detection Using Convolutional Neural Network Method," in *e-Proceeding of Engineering*, 2021.

[9] S. Taliki, S. Serwin, J. Nur and I. C. R. Drajana, " Red Chili Plant Disease Diagnosis Application Using K-Nearest Neighbor Algorithm," *Jurnal Tecnoscienza,* vol. 6, no. 2, pp. 361-373, 2022.

[10] M. Sholikhin and R. Alexandro H, "Disease Classification in Melon Leaf Images Using Convolution Neural Network Algorithm," *Joutica,* vol. 7, no. 1, p. 525, 2022.

[11] A. D. Nurcahyati, R. M. Akbar and S. Zahara, " Classification of Disease Images on Corn Leaves Using Deep Learning with the Convolution Neural Network (CNN) Method," *Jurnal Ilmiah Teknologi Infomasi dan Sains,* vol. 2, no. 2, pp. 43-51, 2022.

[12] I. Supiyani and N. Arifin, " House Number Identification in Digital Images Using Neural Network," *Jurnal Methodika,* vol. 8, no. 1, pp. 18-21, 2022.

[13] M. Sholikhin and R. Alexandro H, " Disease Classification in Melon Leaf Images Using Convolution Neural Network Algorithm," *Joutica,* vol. 7, no. 1, p. 525, 2022.

[14] H. I, S. and H. B. D, " Acne Type Classification Using Convolutional," in *e-Proceeding of Engineering Telkom University*, Bandung, 2021.

[15] Suyanto, Basic and Advanced Machine Learning in Basic and Advanced Machine Learning, Bandung: Pertama., Bandung: Informatika Bandung, 2018.

[16] I. Akil, " Comparison of Neural Network Activation Functions on Time Series Data," *Nusa Mandiri,* vol. 18, no. 1, pp. 78-83, 2023.

[17] S. V, S. D, and S. S, "Implementation of the SoftMax Activation for Reconfigurable Neural Network Hardware Accelerators," *Appl. Sci,* vol. 13, no. 23, p. 12784, 2023.

[18] M. N. Winnarto, M. Mailasari and A. Purnamawati, " Classification of Brain Tumor Types Using Mobilenet V2 Architecture," *Jurnal Simetris,* vol. 13, no. 2, 2022.

[19] F. D. Adhinata, N. A. F. Tanjung, W. Widayat, G. R. Pasfica, and F. R. Satura, "Comparative Study of VGG16 and MobileNetV2 for Masked Face Recognition," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika,* vol. 7, no. 2, p. 230, 2021.

[20] R. Indraswari, R. Rokhana and W. Herulambang, "image classification based on MobileNetV2 network," *Procedia Comput Sci,* vol. 197, p. 198–207, 2022.

[21] N. M. A, W. S. G, T. A. R, and A. S. H, "Performance Comparison of Convolutional Neural Network and MobileNetV2 for Chili Diseases Classification," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi),* vol. 7, no. 4, pp. 940-946, 2023.

[22] R. Kosasih, A. Fahrurozi and D. Riminarsih, " Implementation of Random Forest on Face Recognition Using Isomap Features," *Journal of Computing Engineering, System and Science,* vol. 7, no. 2, p. 459–469, 2022.

[23] C. R. Kotta, D. Paseru and M. Sumampouw, " Implementation of Convolutional Neural Network Method to Detect Diseases in Tomato Leaf Images," *Jurnal Pekommas,* vol. 7, no. 2, pp. 123-132, 2022.

[24] E. Kaban, K. Candra Brata and A. Hendra Brata, " Usability Evaluation Using System Usability Scale (SUS) Method and Discovery Prototyping on PLN Mobile Application (Case Study of PT. PLN)," vol. 11, no. 1, pp. 93-102, 2020.

[25] S. M, R. G, R. D. R, V. K. S, and B. A, "An Empirical Evaluation of Enhanced Performance Softmax Function in Deep Learning," *IEEE Open Access,* vol. 11, p. 34912, 2023.

[26] M. N. Winnarto, M. Mailasari and A. Purnamawati, " Classification of Brain Tumor Types Using Mobilenet V2 Architecture," *Jurnal Simetris,* vol. 13, no. 2, 2022.