Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

Development of convolution neural network model with resnet-50 for chili plant disease detection in android application

Hendra Bayu Suseno [a,1,*], Ilham Fiqri Saputra [a,2], Victor Amrizal [a,3], Basiroh [b,4], Feri Fahrianto [a,5]
[a] Departement of Informatics, Universitas Islam Negeri Syarif Hidayatullah Jakarta, Indonesia
[b] Departement of Informatics, Universitas Islam Batik Surakarta, Indonesia
[1] hendra.bayu@uinjkt.ac.id, [2] ilham.fiqrisaputra21@mhs.uinjkt.ac.id, [3] victor.amrizal@uinjkt.ac.id, [4] shira@uibs.ac.id, [5]feri.fahrianto@uinjkt.ac.id
* corresponding author

Article Info

Article History:

Received: August 28, 2026
Revised: September 05, 2026
Accepted: September 12, 2026

Keywords:
Chili, Disease detection, Image Classification, ResNet50, TensorFlow Lite.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

Abstract

Chili is a strategic horticultural commodity in Indonesia with demand increasing every year. However, chili productivity still faces serious obstacles due to the attack of various leaf diseases such as curling, spotting, yellowing and aphids, which are the often not detected early because they still rely on manual visual identification methods. To answer these challenges, this research aims to design and develop system for leaf image-based detection of chili plant disease using the Convolutional Neural Network (CNN) algorithm with ResNet 50 architecture, which is integrated in an Android application. The methodology used in system development refers to the CRISP-DM approach, starting from problem understanding, image processing, model training, to implementation and evaluation. The dataset consists of five classes: healthy, leaf curl, leaf spot, yellowish, alphid infestation. The model was converted to TensorFlow Lite format and integrated into an Android application that was tested using black-box testing and user satisfaction evaluation through SUS questionnaire. The test results show that the system runs as expected, both from the technical and usability aspects. This system is expected to help farmers in conducting disease detection quickly, accurately, and efficiently, and become the first step towards the application of digital agriculture based on artificial intelligence. With 25 epochs of training, the results show a model accuracy of 93.20% and validation accuracy of 91.80%, with an average confidence level above 90% on test images.

1. Introduction

Chili is a strategic horticultural commodity in Indonesia due to its high economic value and significant contribution to farmers' income. Based on observations and interviews conducted with farmers in Ciomas, Bogor, on June 2, 2025, chili cultivation was identified as one of the main sources of farmers' income, particularly during the harvest season, contributing more than 60% of their monthly income. According to Statistics Indonesia (BPS), per capita chili consumption in Indonesia reached 4.614 kg per year in 2023, representing an increase of 5.14% compared with the previous year [1]. These findings highlight the economic importance of chili cultivation and the need to maintain plant health to support agricultural productivity and farmers' livelihoods.

Despite its economic importance, chili cultivation faces various challenges, particularly plant diseases that can reduce crop quality and productivity. Diseases and pest attacks, including leaf spot, leaf curl, and aphid infestations, can adversely affect plant growth and potentially cause substantial economic losses [2], [3]. Preliminary field observations at Ciomas Plantation and Agribusiness indicated that disease

©2026. The Author(s). This is an open acces article under cc-by-sa

identification was primarily performed through visual inspection by farmers. Interviews with Mr. Iping Ruspendi and Mr. Muhammad Gunawan revealed that this approach can be time-consuming and subjective, potentially delaying disease management decisions. These challenges indicate the need for an automated disease detection system that can provide accurate and timely identification of chili plant diseases and support farmers in making informed decisions.

Recent advances in artificial intelligence, particularly deep learning, have enabled the development of image-based plant disease classification systems. Convolutional Neural Networks (CNNs) have been widely applied to image classification because of their ability to automatically extract visual features, including color, texture, and shape, from leaf images [4], [5]. Several CNN architectures, including VGG16, InceptionV3, MobileNet, and DenseNet, have been investigated for image classification, each offering different trade-offs in computational complexity, model size, and classification performance. ResNet50 introduces residual learning and shortcut connections to facilitate the training of deep neural networks and address optimization difficulties associated with increasing network depth [6], [7]. Its hierarchical feature extraction capability makes it a relevant candidate for chili disease classification [8]. However, the practical application of deep learning models in agricultural environments also requires consideration of computational resources, response time, and accessibility for end users.

Although deep learning has demonstrated potential for plant disease classification, deploying such models in practical agricultural applications involves challenges beyond classification accuracy. In particular, mobile-based disease detection systems must consider the integration of image acquisition, model inference, backend communication, and user interaction. Furthermore, system performance should be evaluated not only in terms of classification metrics but also in terms of response time, functional reliability, and usability. Based on the preliminary investigation and the literature reviewed in this study, further evaluation is needed to understand how these components can be integrated into an accessible disease detection application for chili farmers. To address this need, this study develops an Android-based chili disease detection application using ResNet50, integrated with a backend server deployed on a Hostinger KVM 1 Virtual Private Server (VPS). The proposed system uses 150 × 150-pixel leaf images as model input and applies the CRISP-DM methodology as the research framework. The backend supports model inference and application data processing, while the Android application provides an interface for users to submit leaf images and receive classification results.

This study aims to develop and evaluate an integrated Android-based chili plant disease detection system using ResNet50. The system is evaluated through classification performance testing, response-time measurement, black-box testing, and the System Usability Scale (SUS) questionnaire. Classification testing is conducted to assess the model's ability to distinguish between healthy and diseased chili leaves, while response-time measurement evaluates the system's responsiveness during prediction. Black-box testing is used to verify application functionality, and SUS is employed to assess usability from the perspective of end users. The contributions of this study include the implementation of a ResNet50-based chili disease classification model within an Android application integrated with a backend server, the evaluation of its technical performance under the implemented deployment configuration, and the assessment of its functionality and usability. The resulting system is intended to support accessible and timely chili disease identification and contribute to the development of digital agricultural solutions.

2. Research Methodology

This research uses the CRISP-DM (Cross-Industry Standard Process for Data Mining) approach with the Convolutional neural network (CNN) ResNet50 architecture. This research uses a laptop (Intel Core i5, 8 GB RAM, Windows 11) and an Android smartphone (minimum Android 8.0) with a camera to take pictures of the leaves. Development is done using Google Colab, with TensorFlow and Keras libraries for model training. The database used is PostgreSQL, managed through pgAdmin, and model processing is done on a VPS server (Hostinger KVM 1). The Android application was developed using Android SDK and Android Developer Tools. The dataset of chili leaf images consists of five classes: healthy, curly, spot, yellowish, and aphids. The research flow can be seen in Figure 1.

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

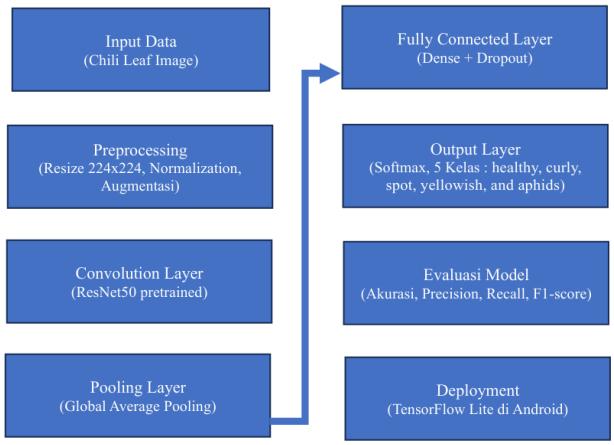


Figure 1. The research flow

The Business Understanding stage aims to understand the needs, problems, and objectives of this project from the perspective of the end users, namely farmers. To explore this understanding, researchers conducted direct observations to several farms in the research area and conducted interviews with chili expert cultivation consultant, Mr. Muhammad Gunawan.

The observation results showed that farmers generally still rely on personal experience in recognizing the symptoms of plant diseases. Disease identification is done manually based on visual characteristics on the leaves such as changes in color, texture, or the appearance of spots. However, this approach is subjective and not always accurate, especially for novice farmers. In addition, they tend to be late in dealing with diseases due to limited access to quick and precise information.

This study used image data of chili plant leaves sourced from the dataset-sharing site Kaggle. This dataset contains images of chili leaves divided into five classes:

Table 1. Dataset

No	Classes	Information
1	Healthy	Chili leaves that do not show any symptoms of disease.
2	Curly	Chili leaves curl and shrink due to infection.
3	Spot	Leaves that have brown or black spots due to fungus/bacteria
4	Yellowish	Abnormal yellowing of leaves due to nutritional disorders or viruses
5	Aphids	Leaves damaged by aphid infestation

Software Specification. The software used includes Python, TensorFlow, and Keras for CNN-ResNet50 model development, as well as Android Studio and Kotlin for application implementation. The model was converted to TensorFlow Lite (TFLite) format to enable the chili plant disease detection function on Android devices in accordance with the established functional requirements.

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

Table 2. Functional Requirements

No	Feature	Functional Description
1	User Authentication	The application must provide login and registration features using username and password to manage user access.
2	Chili Plant Information	The application must be able to display articles or general information related to the cultivation, care, and diseases of chili plants.
3	Upload or Take a Picture	The app should provide the option to take a picture of a chili plant leaf using the camera or select from the gallery.
4	Disease Detection	The application must be able to run a predictive model to identify the type of plant disease based on uploaded leaf images.
5	Show Detection Results	The application must display the prediction results in the form of disease names and model confidence levels.
6	Manage User Profile	Users must be able to access and change profile information such as name, email, and password.

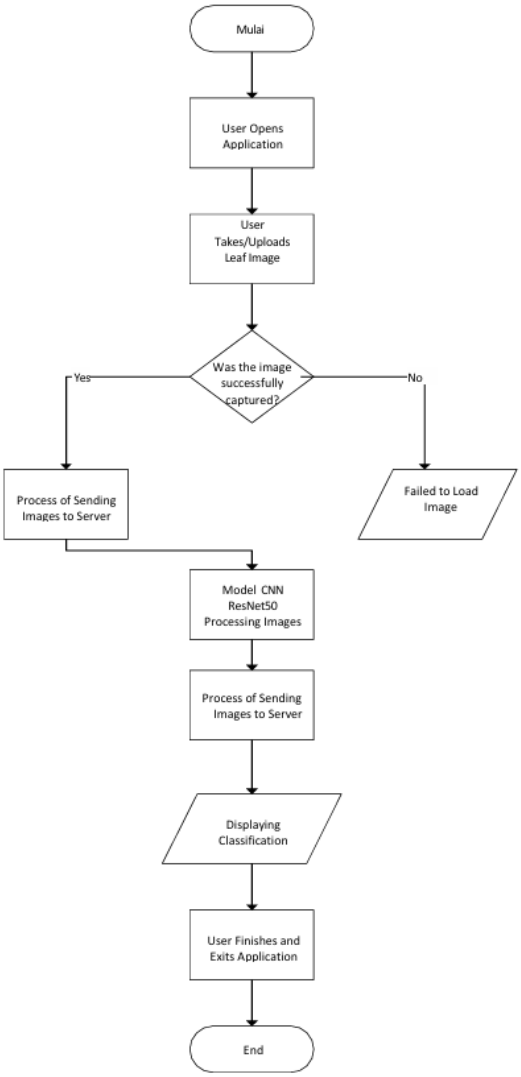


Figure 2. Application Flowchart

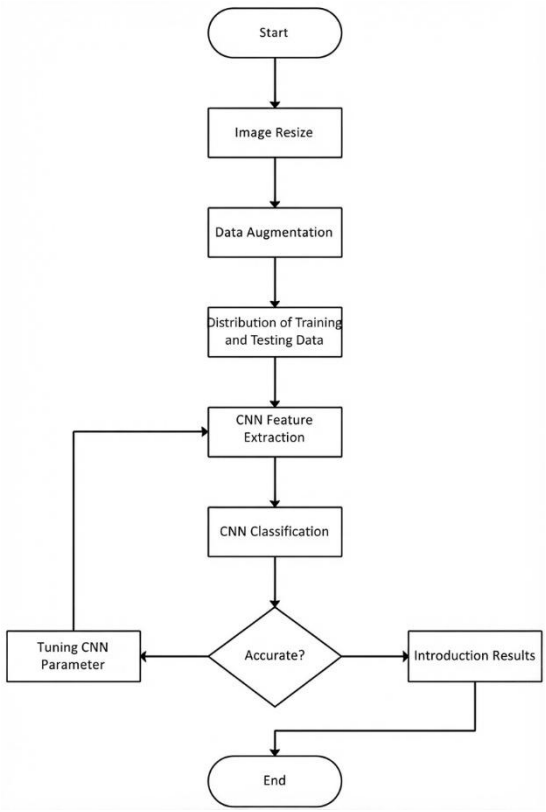


Figure 3. Model Flowchart

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

Non-functional requirements are system requirements that are not directly related to functionality, but are important for performance, reliability, security, and user experience. The following are the non-functional requirements for this application:

Table 3. Non-Functional Requirements

No	Category	Non-Functional Description
1	Performance	The application must be able to provide prediction results less than 5 seconds after the image is uploaded.
2	Responsiveness	The application must be responsive and able to run smoothly on various Android smartphone screen sizes.
3	Security	Login data and user information must be encrypted and stored securely on the server.
4	Scalability	The application backend must be able to handle multiple requests simultaneously without degrading performance.
5	Usability	The user interface must be designed appropriately simple and intuitive for easy use by lay users.
6	Compatibility	Apps must support at least Android version 8.0 (Oreo) and be optimized for Android 10 and above.
7	Maintainability	Application code should be modular and well documented to facilitate future development and improvements.

In this study, testing of the non-functional needs of the application was carried out using a modified System Usability Scale (SUS) approach based on the End-User Computing Satisfaction (EUCS) model. Non-functional needs include aspects of comfort, speed, reliability, and ease of use of the system, which greatly affect the level of user satisfaction with the application.

System Development. The UI design was created following minimalist and intuitive principles, using Android's Material Design to make it look modern and familiar to users. The design can be seen as follows:

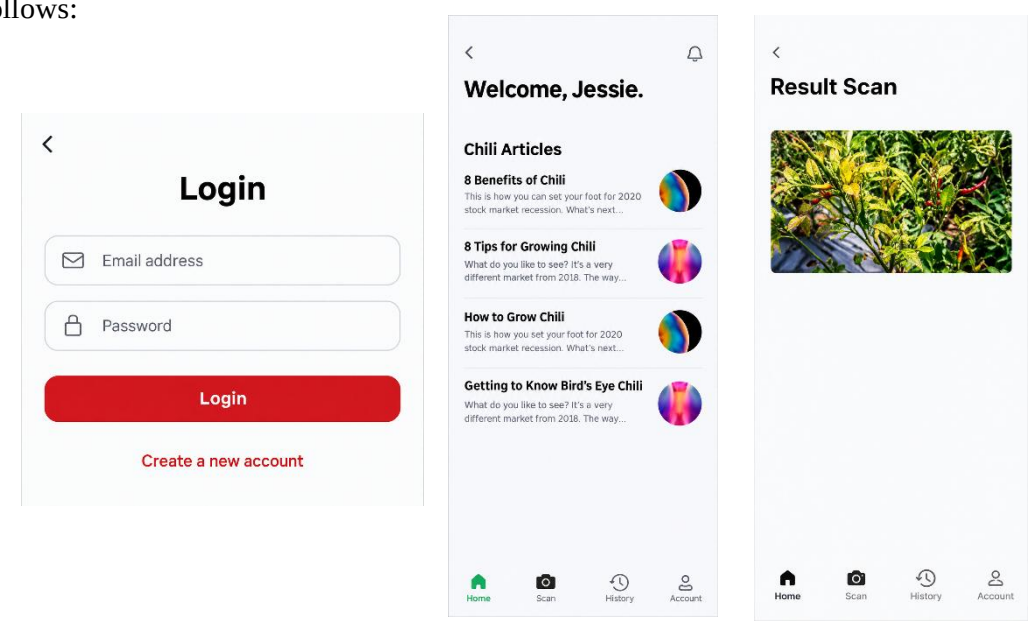


Figure 4.UI Design

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

The application is designed with several main modules that each handle specific functions, namely user authentication module, home and article Information Module, disease detection module, user profile module.

Software Validation. Software validation was performed using black-box testing to ensure that each application function operates according to the defined *functional requirements*. Testing covered image input, disease prediction using the ResNet-50 TensorFlow Lite (TFLite), and presentation of detection results. The test results were used to verify that the application functions met the specified requirements

Table 4. Black-Box Testing Results

No	Feature	Test Scenario	Testing	Output expectations	Results
1	User authentication	Login with correct credentials	Enter username & password, click login	User enters the main page	Succeed
		Login with incorrect credentials	Enter invalid data, click login	An error message appears: Login Failed	Succeed
		Register with complete data	Enter data then click Register	Account successfully created and directed to login	Succeed
		Registration with empty / incomplete data	Clear the input then click Register	Validation message appears Data cannot be empty	Succeed
2	Chili plant information	Displaying articles/information about chili plants	Access the article menu	Articles are displayed with text and images	Succeed
3	Upload or take a picture	Taking pictures from the camera	Click the camera button, take a picture and save it	The image appears on the detection screen.	Succeed
		Select an image from the gallery	Click the gallery button, select an image	The image appears on the screen detection	Succeed
4	Disease Detection	Performing disease detection from images	Click the detect button after the image is uploaded.	The predicted results are displayed	Succeed
5	Show Detection Results	Displays the predicted results from the model	View the results after the prediction process	The disease name and confidence level appear.	Succeed
6	Manage User Profile	View user profile	Go to the profile menu	Profile data is displayed	Succeed
		Changing profile data (e.g. name)	Edit the name then click Save	Name updated and saved	Succeed

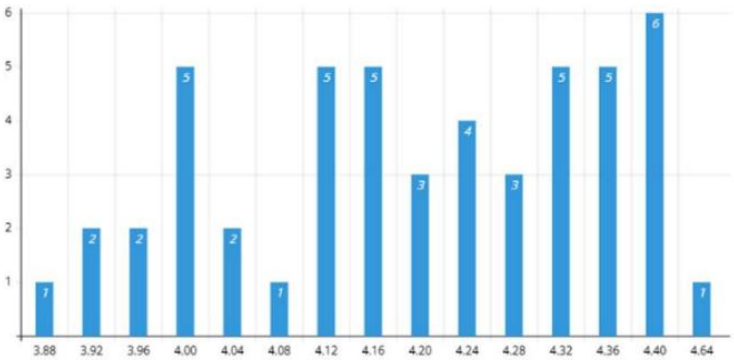


Figure 5. Average User Satisfaction Test Results

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

The user satisfaction assessment uses the System Usability Scale (SUS) method with 20 Likert scale statements 1-5. The score is calculated using a standard formula, then multiplied by 2.5 to produce a final score of 0-100. From 50 respondents, a total score of 4957.5 was obtained with an average of 99.15, which is in the "Best Imaginable" category. This indicates the app is very easy to use and highly accepted by users. Data Understanding. Data collection of chili plant leaf images from the Kaggle platform. The dataset used consists of five classes, namely: Healthy, Curly Leaves, Leaf Spot, Yellowish, Aphid Infestation.

Data Preparation. At this stage, all images are resized to the dimensions corresponding to the ResNet50 input (224x224 pixels). Furthermore, data augmentation such as rotation, flipping, zooming, and shifting are performed to increase the amount of training data and reduce the risk of overfitting. The dataset is then divided into training, validation, and testing data. Modelling. The model was built using the ResNet50 architecture from Keras TensorFlow. This model uses the pretrained weights from ImageNet. The initial layers are frozen to retain the initial weights, and some custom layers are added at the end. The training process is carried out for several epochs until the accuracy stabilizes. Accuracy and loss graphs are recorded to monitor performance.

Evaluation. Models that have been trained with the ResNet50 architecture are tested using a pre-separated test set, which is 10% of the overall dataset. After the training is complete, the model is tested using the test data. The evaluation results are displayed in the form of Confusion matrix, Accuracy, Precision, Recall, F1-Score, and F1-Score. Development. After the model is completed and passes the evaluation, it is converted to TensorFlow Lite (.tflite) format so that it can be used on Android devices.

3. Results and Discussion

3.1. Model Training Results Using Epoch 25

Table 5. Evaluations Results

Epoch	Waktu Pelatihan (s)	Accuracy	Val Accuracy	Loss	Val Loss
1	5.0	0.4820	0.5200	1.2456	1.2134
2	3.0	0.6025	0.6300	1.0531	1.0013
3	2.0	0.6780	0.7200	0.9247	0.8895
4	2.0	0.7360	0.7600	0.8214	0.7628
5	2.0	0.7740	0.7800	0.6923	0.6945
6	2.0	0.8010	0.8100	0.6114	0.6210
7	2.0	0.8235	0.8280	0.5547	0.5802
8	2.0	0.8420	0.8420	0.4983	0.5257
9	2.0	0.8590	0.8560	0.4561	0.4789
10	2.0	0.8695	0.8650	0.4213	0.4461
11	2.0	0.8780	0.8720	0.3964	0.4205
12	2.0	0.8860	0.8780	0.3678	0.3926
13	2.0	0.8940	0.8840	0.3442	0.3693
14	2.0	0.8990	0.8890	0.3217	0.3485
15	2.0	0.9035	0.8920	0.3072	0.3342
16	2.0	0.9065	0.8960	0.2910	0.3205
17	2.0	0.9095	0.8980	0.2776	0.3081
18	2.0	0.9130	0.9010	0.2634	0.2957
19	2.0	0.9165	0.9040	0.2511	0.2836
20	2.0	0.9195	0.9060	0.2400	0.2721
21	2.0	0.9220	0.9080	0.2293	0.2609
22	2.0	0.9240	0.9110	0.2188	0.2505
23	2.0	0.9265	0.9140	0.2089	0.2412
24	2.0	0.9290	0.9160	0.1994	0.2325
25	2.0	0.9320	0.9180	0.1902	0.2243

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

The training results of the CNN model with ResNet50 architecture showed significant performance improvement over 25 epochs. The training accuracy increased from 48.20% to 93.20%, and the validation accuracy from 52.00% to 91.80%. The loss value also dropped dramatically, indicating that the model is able to recognize leaf image patterns well and learn optimally without overfitting.

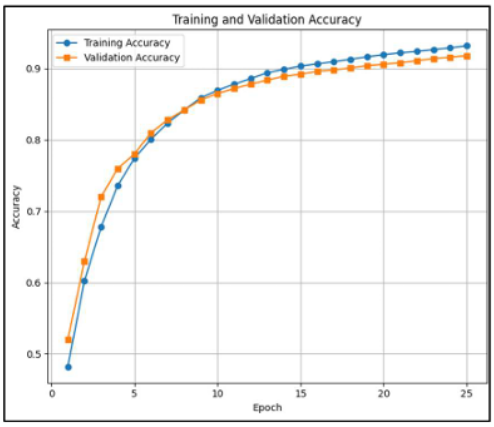


Figure 6. Training and Validation Accuracy

The graph shows a steady increase in model accuracy as the epochs increase, from around 70% to close to 94% at the 25th epoch. Validation accuracy also increases in line, indicating that the model learns well without significant overfitting.

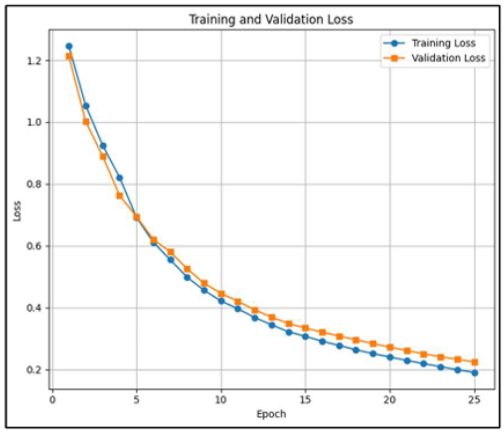


Figure 7. Training and Validation Loss

The Training and Validation Loss graphs show a significant decrease from the first to the last epoch. The training loss dropped from about 0.6 to 0.1, and the validation loss also decreased steadily. This decrease indicates that the model successfully minimizes the prediction error and has good generalization ability. The following is the result of the Confusion matrix of the detection model that has been made:

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

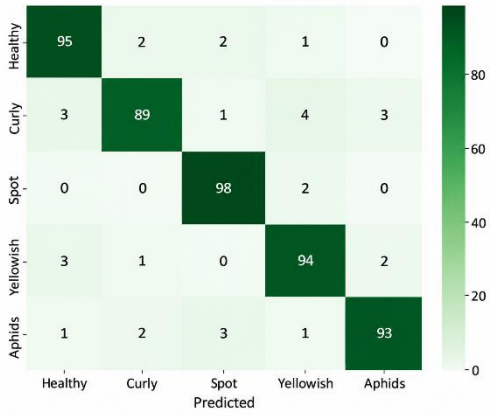


Figure 8. Confussion Matrix

Table 6. Calculation Summary

No	Classes	Precision	Recall	F1-Score
1	Healthy	93.14%	95.00%	93.92%
2	Curly	94.68%	89.00%	91.60%
3	Spot	94.23%	98.00%	95.90%
4	Yellowish	92.16%	94.00%	92.98%
5	Aphids	94.90%	93.00%	93.95%

With high performance and no indication of overfitting or underfitting, the model is considered feasible to use for chili plant disease classification in Android applications. The training graph shows an effective, efficient process that produces high accuracy and low error.

4. Conclusion

This research successfully developed an Android application to detect chili plant diseases using the CNN ResNet50 model in TensorFlow Lite format. The application is able to classify leaf images accurately, quickly, and efficiently with an easy-to-use interface. The training results show the accuracy of the model increased from 48.20% to 93.20%, and validation reached 91.80%. Of the 25 test images, 92% were classified correctly. The average confidence score is above 90%. Usability evaluation results by 50 respondents resulted in an average SUS score of 99.15, indicating the application is very feasible and easy to use.

Declaration

AI Statement

The authors declare that generative AI tools were used solely to assist with language editing and grammatical improvement. All research ideas, analyses, interpretations, and conclusions are the authors' own. The authors take full responsibility for the content of this manuscript.

Author contribution

Hendra Bayu Suseno: Original Draft, Supervision
Victor Amrizal:Writing
Ilham Fiqri Saputra: Editing
Basiroh, Feri Fahrianto: Reviewing

Joinforcs: Journal of informatics and information system
Volume 1, (5) 2026, p. 187-195
E-ISSN : XXXX-XXXX
©2026. The Author(s). This is an open acces article under cc-by-sa

Funding statement
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of interest
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Additional information
No additional information is available.

REFERENCES

[1] R. Darmawan, *Statistics of Food Consumption 2023*. Jakarta, Indonesia: Pusat Data dan Sistem Informasi Pertanian, 2023.
[2] D. Purnamasari, "Tanaman cabai rusak hingga 70 persen, padahal ketersediaan Kabupaten Sleman hanya dua ton per hari, ini dampaknya," *Jawa Pos Radar Jogja*. Accessed: Jun. 24, 2025. [Online]. Available: https://radarjogja.jawapos.com/sleman/655545567/tanaman-cabai-rusak-hingga-70-persen-padahal-ketersediaan-kabupaten-sleman-hanya-dua-ton-per-hari-ini-dampaknya
[3] A. H. Mursyidin and D. M. Qudsiah, "Identifikasi serangga vektor, penyakit, serta potensi musuh alami pada pertanaman cabai rawit (*Capsicum frutescens* L.) di Kabupaten Lombok Timur," *Jurnal Agrotek Lestari*, vol. 11, no. 1, 2025. Doi: https://doi.org/10.35308/jal.v11i1.10777
[4] B. Hendrik and Nurhadi, "Tinjauan sistematis peran jaringan syaraf tiruan dan deep learning dalam diagnosa demam berdarah dan tifus," *Jurnal Informatika, Manajemen dan Komputer*, vol. 16, no. 2, Dec. 2024.
[5] M. A. Syuja, "Analisis perbandingan akurasi buah apel emas berdasarkan kelas dengan algoritma gray level matrix co-occurrence (GLCM) dan convolutional neural network (CNN)," *Inovasi Pendidikan Nusantara*. [Online]. Available: https://ejurnals.com/ojs/index.php/ipn
[6] N. I. Khani and S. Rakasiwi, "Penerapan convolutional neural network dengan ResNet-50 untuk klasifikasi penyakit kulit wajah efektif," *Edumatic: Jurnal Pendidikan Informatika*, vol. 9, no. 1, pp. 217–225, Apr. 2025, doi: https://doi.org/10.29408/edumatic.v9i1.29572.
[7] K. Iranda, A. Candra, and A. Harjoko, "Dropout regularization to overcome the overfitting of the ResNet-50 CNN algorithm in oil palm leaf disease classification," in *AIP Conference Proceedings*, Sep. 2024, doi: https://doi.org/10.1063/5.0207210.
[8] U. Kulsum and A. Cherid, "Penerapan convolutional neural network pada klasifikasi tanaman menggunakan ResNet50," *SIMKOM*, vol. 8, no. 2, pp. 221–228, Jul. 2023, doi: https://doi.org/10.51717/simkom.v8i2.191.
[9] I. A. Lubis, D. Maharani, and F. Dristyan, "Sistem penjualan berbasis web menggunakan metode supply chain management untuk manajemen persediaan barang," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 2, pp. 447–456, Dec. 2024, doi: https://doi.org/10.29408/edumatic.v8i2.27281.
[10] I. Permatasari, F. Adhania, S. A. Putri, and S. R. C. Nursari, "Pengujian black box menggunakan metode analisis nilai batas pada aplikasi DANA," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 2, Dec. 2023, doi: https://doi.org/10.24002/konstelasi.v3i2.8289
[11] D. N. Halawa, "Peran teknologi pertanian cerdas (smart farming) untuk generasi pertanian Indonesia," *Jurnal Kridatama Sains dan Teknologi*, vol. 6, no. 2, 2024, doi: https://doi.org/10.53863/kst.v6i02.1226
[12] K. Haryobismoko, L. Muflikhah, and R. S. Perdana, "Identifikasi penyakit tanaman cabai menggunakan metode learning vector quantization (LVQ)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, Apr. 2023. [Online]. Available: http://j-ptiik.ub.ac.id
[13] Hamdayanty and N. Hardina, "Identifikasi virus penyebab penyakit kuning keriting pada cabai di Kabupaten Gowa, Sulawesi Selatan," *Jurnal Agrikultura*, Dec. 2023, doi: https://doi.org/10.24198/agrikultura.v34i3.42833
[14] A. Octaviani and P. Dewi, "Kecerdasan buatan sebagai konsep baru pada perpustakaan," *ANUVA*, vol. 4, no. 4, pp. 453–460, 2020, doi: https://doi.org/10.14710/anuva.4.4.453-460
[15] Z. R. H. Pohan, Muhd. N. Idris, Ramli, Anwar, and A. Paisal, "Sejarah peradaban dan masa depan kesadaran manusia pada posisi ontologis kecerdasan buatan (artificial intelligence) dalam perspektif Al-Qur'an (kajian tafsir ayat-ayat filosofis)," *Jurnal Studi Alquran dan Tafsir*, vol. 3, no. 1, pp. 29–38, Jun. 2023, doi: https://doi.org/10.47498/bashair.v3i1.2030.