E-ISSN : 2988-585X (Online)

Journal of Elektronik Sistem InformasI (JESII)

Volume 2 No 2 Desember 2024

DOI : 10.31848/jesii.xxxx.xxxx

The Implementation of Machine Learning for Disease Detection in Tomato Plants using Convolutional Neural Networks

Faisal Aziz[1], Nana Suryana[2]

[1,2]Department of Information System, Univ. Kebangsaan Republik Indonesia, Indonesia

Article Info

Article history:

Received Dec 16, 24
Revised Dec 30, 24
Accepted Dec 31, 24

Keywords:

Convolutional Neural Network
Machine Learning
Tomato Plant Diseases

ABSTRACT

Diseases in tomato plants can be highly detrimental to tomato farmers, with common afflictions such as begomovirus, blight, and spider mites posing significant challenges. The implementation of machine learning offers a promising solution to address these issues and mitigate the financial losses caused by such diseases. This study aims to evaluate the effectiveness of machine learning in detecting plant diseases using Convolutional Neural Networks (CNN). The data used in this implementation was obtained from public datasets available on Kaggle and real-time data collected directly from tomato farms in Kadudampit, Sukabumi Regency. The Kaggle dataset contains 4,800 images categorized into three classes: begomovirus, blight, and spider mites. Meanwhile, the real-time dataset comprises 450 images, also divided into the same three classes. The performance of the machine learning model was tested using different datasets, with accuracy measured through a confusion matrix. The results showed that the machine learning model trained on the public dataset achieved the highest accuracy of 97%. The model trained on a combination of the public and real-time datasets achieved an accuracy of 94%, while the model trained solely on the real-time dataset achieved an accuracy of 80%. A machine learning model is considered effective if its accuracy exceeds 75%. Therefore, based on the three tests conducted, it can be concluded that the machine learning models demonstrated a good level of accuracy in detecting diseases in tomato plants.

Corresponding Author :

Faisal Aziz,
Information System Department, Faculty of Computer Science and Information Systems, Univ. Kebangsaan Republik Indonesia.
Jln. Terusan Halimun No.37 (Pelajar Pejuang 45) Bandung, Jawa Barat, Indonesia. 40614
Email: faisalaziz@gmail.com

1. INTRODUCTION

With the advancement of technology, particularly in the field of Artificial Intelligence (AI), various machine learning methods have been developed to address a wide range of issues in agriculture. One prominent method is Convolutional Neural Network (CNN). CNN, a type of deep learning, is known for its effectiveness in image analysis and pattern recognition. In the context of plant disease detection, CNN holds great potential for identifying disease symptoms on plant leaves through images with high accuracy [1].

This study focuses on tomato plants, specifically their leaves. Leaves are among the most responsive parts of a plant in showing early symptoms of disease. These symptoms may include discoloration, the appearance of spots, or deformation of the leaf structure. Early identification of these symptoms is crucial to ensure timely and appropriate intervention before the disease spreads and infects the entire plant. This research employs machine learning technology, with Convolutional Neural Network (CNN) as the primary method. CNN excel at visual data analysis such as image classification and object detection, making them a key tool for visual pattern-based decision making, especially in the medical, surveillance, and autonomous vehicle fields [2]. In this study, CNN is utilized to analyze tomato leaf images to automatically detect diseases.

The main challenge of this research lies in implementing a CNN model capable of detecting diseases in tomato plants with high accuracy. This challenge is further complicated by variations in lighting conditions, image angles, and differing visual characteristics of diseases. To overcome these issues, this study employs advanced techniques such as data augmentation and transfer learning, which have proven to improve performance in visual recognition tasks [3] [4]. Data collection involves capturing tomato leaf images in the field and utilizing public datasets, which are then processed and labeled for training the CNN model [5]. The performance of the developed model will be measured using metrics such as accuracy, precision, recall, and F1-score [6].

Unlike previous studies, this study uses a combination of public data from Kaggle and real-time data collected directly from agricultural locations in Kadudampit, Sukabumi. This approach provides a unique contribution because real-time data reflects local conditions more accurately than public data, which is often generic and may not reflect conditions on the ground. With proper implementation, this technology can significantly contribute to early disease detection, assisting farmers in better disease management, and ultimately improving crop yields. Furthermore, this study aims to promote the broader adoption of smart agricultural technologies in modern farming practices.

2. METHOD

Convolutional Neural Network (CNN) is a type of deep learning algorithm capable of training models with a large number of parameters, often reaching millions, using 2D images as input. This approach is specifically designed to process and analyze visual data such as images. CNN employs filters or kernels to extract information from images, generating the desired output. This process enables CNN to effectively capture and understand complex patterns within images, making it one of the most powerful tools in various applications, including object recognition, image classification, and pattern detection [7] [8].

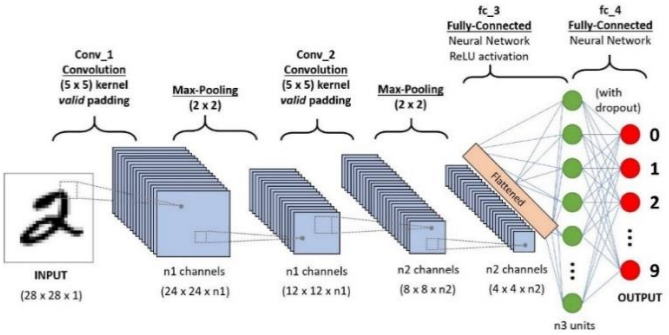


Figure 1. Convolutional Neural Network

In general, there are three main layers in Convolutional Neural Networks (CNN): convolution layers, pooling layers, and fully connected layers [9] [10].

a. Convolution Layers

Convolution Layers are the core components of Convolutional Neural Networks (CNN) responsible for processing the dimensions of an image, such as its width, height, and depth, along with the kernel. Their primary function is to act as filters and generate feature maps during image input processing [11].

By applying kernels (small matrices) to the input image, convolution layers extract specific features, such as edges, corners, and textures, by sliding these filters over the image. The output, known as the feature map, represents the presence and intensity of these features, enabling the model to focus on significant patterns in the data while ignoring irrelevant details.

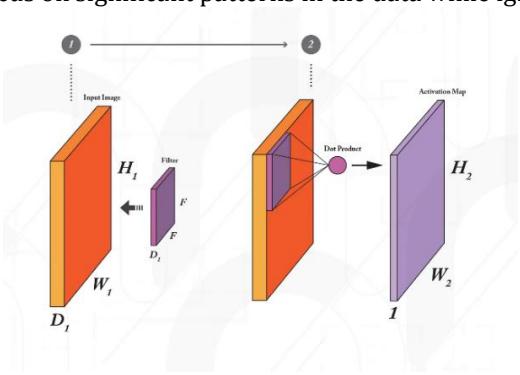


Figure 2. Convolution Layers

b. Pooling Layers

The pooling layer plays an important role in reducing the spatial dimensions of data to decrease the number of parameters and computations required. Additionally, this layer helps prevent overfitting, where the model achieves high accuracy in predicting training data but fails to recognize data outside the training sample. There are various commonly used pooling methods, such as max pooling and average pooling. In max pooling, the maximum value is taken from a specific region, while in average pooling, the average value is taken [12].

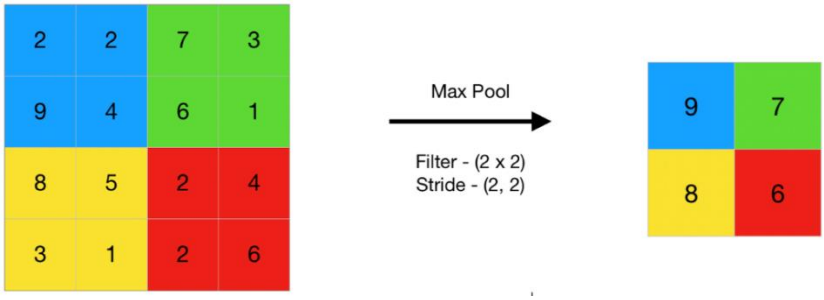


Figure 3. Pooling Layers

c. Fully Connected Layers

The fully connected layer is the final layer in the CNN architecture. Each neuron in this layer is connected to every neuron in the preceding or succeeding layer. This layer is often used as the output layer or the classifier in CNNs [13].

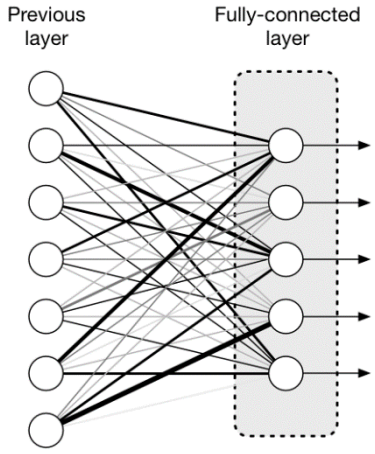


Figure 4. Fully Connected Layers

To evaluate machine learning, a confusion matrix is used to measure the accuracy achieved by the model using three different datasets. Confusion matrix evaluates the performance of a classification model by summarizing predictions vs. actual results. Used in supervised learning, it identifies accuracy, error, and priority metrics for model optimization. [14]. This matrix presents a comparison between the predictions made by the model and the actual values from the test data in a tabular form. For binary classification problems, the confusion matrix consists of four main components: True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). True Positive refers to cases where the model correctly predicts a positive outcome, while False Negative represents cases where the model fails to detect an actual positive value. False Positive occurs when the model incorrectly predicts a negative outcome as positive, and True Negative refers to cases where the model correctly predicts a negative value [15].

Here is the basic structure of a confusion matrix:

Table 1. Basic structure of confusion matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	False Positive (FP)

Here is the explanation for each component:

1. True Positive (TP): The case where the model predicts positive, and the actual value is also positive.
2. False Negative (FN): The case where the model predicts negative, but the actual value is positive.
3. False Positive (FP): The case where the model predicts positive, but the actual value is negative.
4. True Negative (TN): The case where the model predicts negative, and the actual value is also negative.

From the confusion matrix, several evaluation metrics can be calculated, including:

1. Accuracy: The percentage of correct predictions from the overall predictions is referred to as Accuracy.
   Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

2. Precision: The proportion of correct predictions
   Precision = $\frac{TP}{TP+FP}$

3. Recall (Sensitivity): The proportion of positive cases correctly detected by the model
   Recall = $\frac{TP}{TP+FN}$

4. F1 Score**:** Harmonic mean from precision and recall.
   F1 Score = 2 x $\frac{Precision\ x\ Recall}{Precision+Recall}$

SDLC (Software Development Life Cycle) is an organized framework for the software development process, which has evolved from the Waterfall model to Agile and DevOps. Modern approaches emphasize speed, adaptability, automation, and the ability to respond quickly to change [16]. In the machine learning implementation process, SDLC is used as the method for software development. Here are the stages of SDLC:

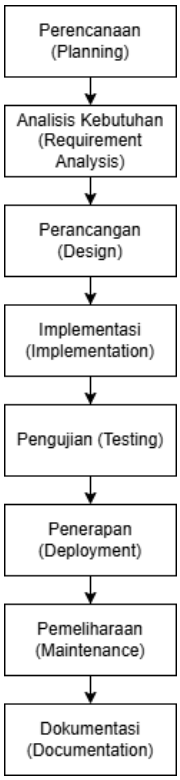


Figure 5. SDLC Stages

1. The first stage in the implementation process of this machine learning model is the planning stage. In this stage, the objective of this research will be identified, which is to implement a machine learning model to detect diseases in tomato plants using CNN [17].

2. The second stage in the machine learning implementation process is the requirements analysis. Functional requirements describe the services that a software system must provide, while non-functional requirements are concerned with the quality and constraints of the system [18].
    a. Functional Requirements
        i. Detection & Classification: The model should detect and classify three types of diseases from leaf images (begomovirus, blight, and spider mite disease categories).
        ii. Training & Evaluation: The system is trained using public and real-time datasets, evaluated using accuracy, precision, and F1-score.
        iii. Platform: Implementation can be done on Google Collab.
    b. Non-Functional Requirements
        i. Accuracy >75%.
        ii. Use data augmentation and support the addition of new disease classes.
        iii. Consistent results despite varying image conditions.
    c. Data Requirements
        i. Dataset Public (4,800 images) and field (450 images) datasets with disease category labels.
        ii. Augmentation techniques to increase data diversity.
3. The method used in this research is CNN with a sequential model. The model-building process involves three layers in accordance with the CNN architecture, which are the convolution layers, pooling layers, and fully connected layers.
4. In the implementation process of creating this machine learning model, there are several stages that need to be performed. To facilitate the creation process, a flowchart has been created as follows:

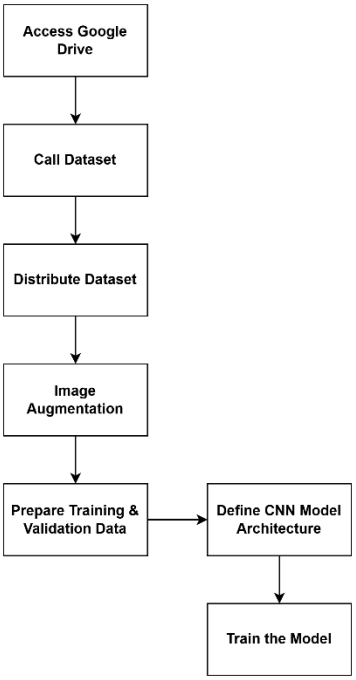


Figure 6. Implementation Process

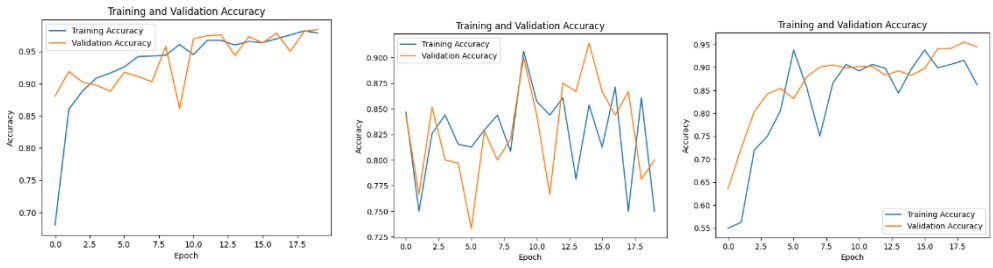
5. In the testing stage of the machine learning process, testing will be conducted by inputting images of diseases in tomato plants with leaf images, which will later be detected for diseases by the machine learning model. Evaluation will then be performed using confusion matrix, F1 Score, precision, and recall.
6. The next step in the implementation process of the machine learning model for detecting diseases in tomato plants is deployment. The machine learning model can be used on the Google Collab platform by downloading the pre-uploaded model from Kaggle [19].
7. The maintenance stage is the 7th phase in the machine learning implementation process. During this phase, updates or additions to the data will be made if there are new disease classes to be added. To add a class, new data can be uploaded to Google Drive, which is connected to the model. Below is an example of adding a disease class.
8. Documentation is the final stage in the process of creating a machine learning model for detecting diseases in tomato plants. In this phase, the machine learning documentation will be presented and stored on the Kaggle website.

3. RESULT AND DISCUSSION

The implementation of machine learning using three different datasets will result in varying accuracies, as this is influenced by the quality and quantity of the datasets. However, from this, the impact and quality of the datasets can be observed, which will affect the accuracy of the machine learning model [20]. For the evaluation using these three datasets, calculations will include training and validation accuracy, F1 confidence curve, precision confidence curve, recall confidence curve, precision-recall curve, and the confusion matrix.

A. Training and validation accuracy

The accuracy of the training data and validation data will be presented in the form of a plot graph, allowing the differences between each dataset used to be clearly visualized.

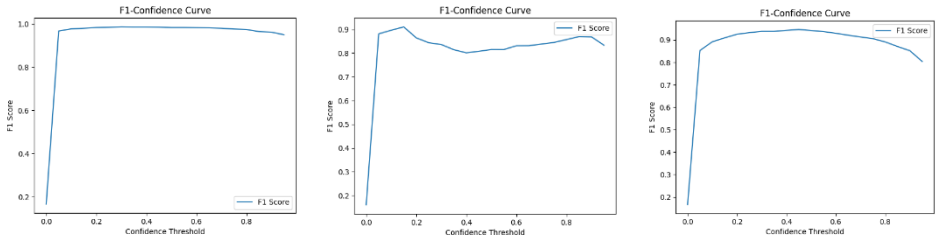


Public Data | Realtime Data | Realtime & Public Data

From the graph, it can be seen that the accuracy of the training and validation data on the public dataset appears more consistent and stable. This is due to the public dataset being well-organized and having minimal noise. In contrast, the real-time dataset tends to show a more irregular graph. This is because the real-time dataset contains more noise, which leads to fluctuating training and validation accuracy, making the graph appear less stable.

B. F1 Confidence Curve

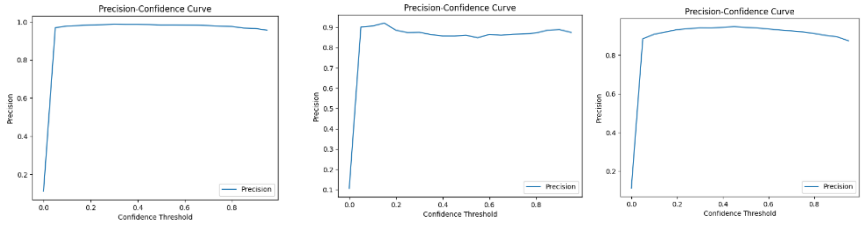
The F1 confidence curve will be presented in the form of a graph for each dataset to highlight the differences between the datasets used.



Public Data | Realtime Data | Realtime & Public Data

C. Precision Confidence Curve

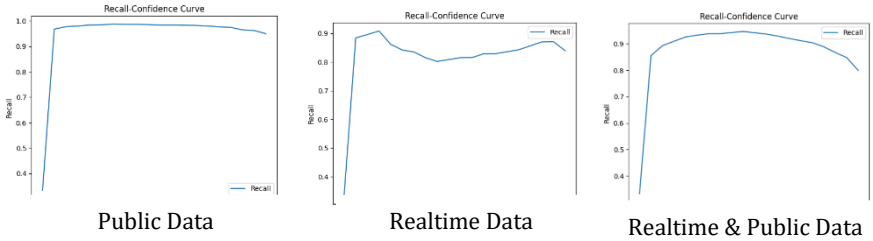
The Precision Confidence Curve will be presented in the form of a graph for each dataset to identify the differences between the datasets used.



Public Data | Realtime Data | Realtime & Public Data

D. Recall Confidence Curve

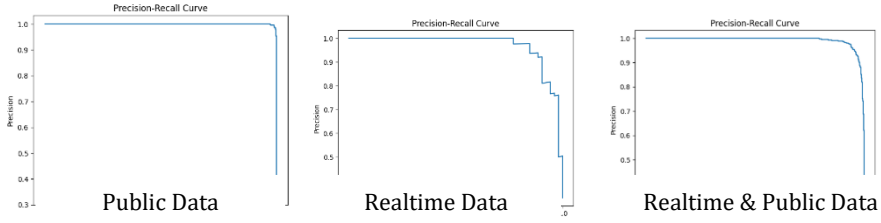
The Recall Confidence Curve will be presented in the form of a graph for



Public Data | Realtime Data | Realtime & Public Data

E. Precision Recall Curve

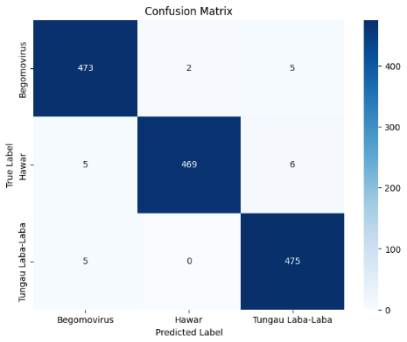
The Precision-Recall Curve will be presented in the form of a graph for each dataset to highlight the differences between the datasets used.



Public Data | Realtime Data | Realtime & Public Data

F. Confusion Matrix

The Confusion Matrix will calculate the accuracy of the model using different datasets. The model must achieve an accuracy of >75% to be considered good.



Data Publik

Begomovirus

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{473}{473+2}$ = 0.9958

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{473}{473+5}$ = 0.9895

F1 Score = 2 x $\frac{0.9958\ x\ 0.9895}{0.9958\ +\ 0.9895}$ = 0.9926

Hawar

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{469}{469+11}$ = 0.9771

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{469}{469+0}$ = 1.0

F1 Score = 2 x $\frac{0.9771\ x\ 1.0}{0.9771\ +1.0}$ = 0.9884

Tungau Laba-Laba

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{475}{471+5}$ = 0.9896

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{475}{475+0}$ = 1.0

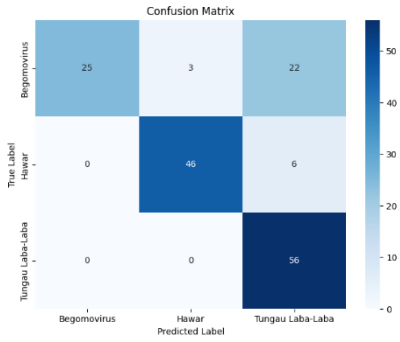
F1 Score = 2 x $\frac{0.9896\ x\ 1.0}{0.9896\ +1.0}$ = 0.9948

To measure the overall accuracy of the model, the following formula is used for calculation:

Accuracy = $\frac{Number\ of\ correct\ prediction}{Total\ Number\ of\ Data}$

Accuracy = $\frac{473+469+475}{1440}$

Accuracy = <u>0.9840 or 98%</u>



Data Realtime

Begomovirus

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{25}{25+3}$ = 0.8929

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{25}{25+22}$ = 0.5319

F1 Score = 2 x $\frac{0.8929x\ 0.8929}{0.5319\ +\ 0.5319}$ = 0.6667

Hawar

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{46}{46+6}$ = 0.8846

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{46}{46+0}$ = 1.0

F1 Score = 2 x $\frac{0.8846\ x\ 1.0}{0.8846\ +1.0}$ = 0.9387

Tungau Laba-laba

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{56}{56+0}$ = 1.0

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{56}{56+0}$ = 1.0

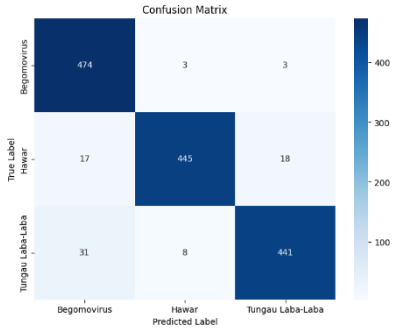
F1 Score = 2 x $\frac{1.0\ x\ 1.0}{1.0\ +1.0}$ = 1.0

To measure the overall accuracy of the model, the following formula is used for calculation:

Accuracy = $\frac{Number\ of\ correct\ prediction}{Total\ Number\ of\ Data}$

Accuracy = $\frac{25+46+56}{158}$

Accuracy = <u>0.8037 or 80%</u>



Data Realtime & publik

Begomovirus

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{474}{474+20}$ = 0.9595

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{445}{445+35}$ = 0.9875

F1 Score = 2 x $\frac{0.9595\ x\ 0.9875}{0.9595 + 0.9875}$ = 0.9733

Hawar

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{445}{445+21}$ = 0.9554

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{445}{445+35}$ = 0,9271

F1 Score = 2 x $\frac{0.9554\ x\ 0,9271}{0.9554 + 0,9271}$ = 0.9410

Tungau Laba-laba

Precision = $\frac{TP}{TP+FP}$

Precision = $\frac{441}{441+21}$ = 0.9545

Recall = $\frac{TP}{TP+FN}$

Recall = $\frac{441}{441+39}$ = 0,9187

F1 Score = 2 x $\frac{0.9545\ x\ 0,9187}{0.9545 + 0,9187}$ = 0.9362

To measure the overall accuracy of the model, the following formula is used for calculation:

Accuracy = $\frac{Number\ of\ correct\ prediction}{Total\ Number\ of\ Data}$

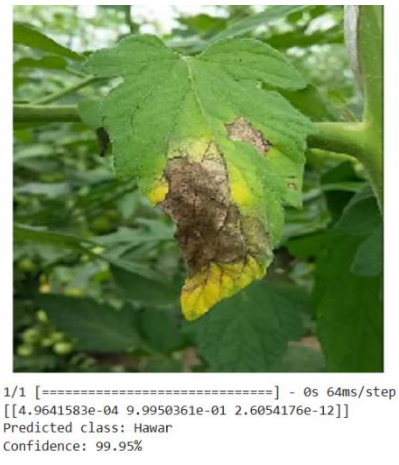
Accuracy = $\frac{474+445+441}{1440}$

Accuracy = 0.9444 or 94%

In addition, testing was conducted on tomato leaves affected by diseases using three different datasets.

1. Testing with the Public Dataset.

Begomovirus



Blight

```
1/1 [==============================] - 0s 59ms/step
[[2.8741410e-05 8.6444797e-06 9.9996257e-01]]
Predicted class: Tungau Laba-laba
Con
```

Spider mites

2. Testing with the Realtime Dataset.

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 137ms/step
[[0.8937802  0.08167537 0.02454451]]
Predicted class: Begomovirus
Confidence: 89.38%
```

Begomovirus

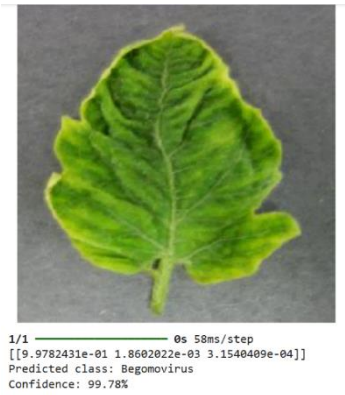
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step
[[8.6751173e-04 3.5872194e-04 9.9877375e-01]]
Predicted class: Tungau Laba-laba
Confidence: 99.88%
```

Spider mites

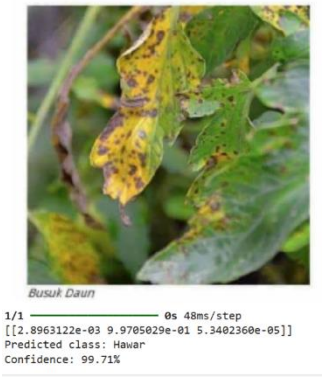
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
[[0.05086412 0.9476127  0.00152308]]
Predicted class: Hawar
Confidence: 94.76%
```

Blight

3. Testing with the Public and Realtime Dataset.



Begomovirus



Blight

Spider Mites

4. CONCLUSION

The conclusion drawn from the implementation of a machine learning model for detecting diseases in tomato plants using a CNN architecture is that the model successfully achieved an average accuracy of 90.33%. A machine learning model is considered highly effective if its accuracy exceeds 75%. The results show that the highest accuracy, 97%, was obtained using the Kaggle public dataset, while a combination of public and real-time datasets yielded an accuracy of 94%. Meanwhile, using only the real-time dataset resulted in an accuracy of 80%. Moreover, to achieve optimal accuracy, it is crucial to ensure that the dataset contains clear, specific images free from interference by other objects, as these factors can significantly affect the model's performance in accurately detecting diseases. Thus, the quality and cleanliness of the data play a critical role in the success of machine learning applications in tomato plant disease detection. There are also suggestions for future research by integrating CNN models with Internet of Things (IoT) devices, such as drones or field cameras, to detect diseases in real-time and accelerate interventions in the field.

REFERENCES

[1] K. C. H and L. Hernando, "Implementasi Sistem Pendeteksi Penyakit Pada Daun Singkong Dan Daun Cabai Berbasis Machine Learning," *Jurnal Quancom,* vol. 1, no. 2, pp. 1-5, 2023.

[2] H. Taherdoost, "Deep Learning and Neural Networks: Decision-Making Implications," *Symmetry,* vol. 15, pp. 1-22, 2023.

[3] M. Liu, "Analysis of Image Data Augmentation in Smart Agriculture," *Highlights in Science, Engineering and Technology,* vol. 85, pp. 750-756, 2024.

[4] W. Shafk, A. Tufail, C. D. S. Liyanage and R. A. A. H. M. Apong, "Using transfer learning-based plant disease classifcation and detection for sustainable agriculture," *BMC Plant Biology,* vol. 24, no. 136, pp. 1-19, 2024.

[5] K. ZOU, Y. SHAN, X. ZHAO, D. C. RAN and X. CHE, "A Deep Learning Image Augmentation Method for Field Agriculture," *IEEE,* vol. 12, pp. 37432 - 37442, 2024.

[6] M. Ali, T. Hariyati, M. Y. Pratiwi and S. Afifah, "Metodologi Penelitian Kuantitatif Dan Penerapan Nya Dalam Penelitian," *Education Journal : Penelitian Ibnu Rusyd Kotabumi,* vol. 2, no. 2, 2022.

[7] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang and C. Xu, "arXiv:2107.06263," *CMT: Convolutional Neural Networks Meet Vision Transformers,* 2021.

[8] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data,* vol. 8, no. 53, 2021.

[9] R. Ardianto and S. K. Wibisono, "Analisis Deep LearningMetode Convolutional Neural NetworkDalam Klasifikasi Varietas Gandum," *JURNAL KOLABORATIF SAINS,* vol. 6, no. 12, pp. 2082-2092, 2023.

[10] Purwono, A. Ma'arif, W. Rahmaniar, H. I. K. Fathurrahman, A. Z. K. Frisky and Q. M. u. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *IJRCS : International Journal of Robotics and Control Systems ,* vol. 2, no. 4, pp. 739-748, 2023.

[11] M. AZWAN, "ANALISIS RESIDUAL NETWORK (RESNET) UNTUK KLASIFIKASI JENIS PENYAKIT PADA TANAMAN CABAI MELALUI CITRA DAUN," UNIVERSITAS MEDAN AREA, MEDAN, 2023.

[12] E. V. Tjahjadi, B. Santoso and Serwin, "Klasifikasi Malware Menggunakan Teknik Machine Learning," *Banthayo Lo Komputer,* vol. 2, no. 1, pp. 60-70, 2023.

[13] I. Cholissodin and A. A. Soebroto, Buku Ajar AI, Machine Learning & Deep Learning, Malang: Fakultas Ilmu Komputer (FILKOM), Universitas Brawijaya (UB), Malang., 2019.

[14] S. Sathyanarayanan and B. R. Tantri, "Confusion Matrix-Based Performance Evalua□on Metrics," *African Journal of Biomedical Research,* vol. 27, no. 4, pp. 4023-4031 , 2024.

[15] M. Heydarian, T. E. Doyle and R. Samavi, "MLCM: Multi-Label Confusion Matrix," *IEEE,* vol. 10, pp. 19083-19095, 2022.

[16] R. Ranawana and A. S. Karunananda, "An Agile Software Development Life Cycle Model for Machine Learning Application Development," *IEEE,* pp. 1-6, 2021.

[17] A. Setiadi, R. A. Ridwan and N. R. Rizqullah, "Sistem Informasi Booking Futsal Menggunakan Metode Agile SDLC Pada KAO Futsal," *Journal Sensi: Strategic of Education in Information System,* vol. 7, pp. 1-12, 2021.

[18] C. Dongmo, "A Review of Non-Functional Requirements Analysis Throughout the SDLC," *Computers,* vol. 13, no. 308, pp. 1-20, 2024.

[19] K. Banachewicz and L. Massaron, The Kaggle Book: Data analysis and machine learning for competitive data science, Birmingham: Packt Publishing Ltd, 2022.

[20] L. Syafa'ah, Z. Zulfatman, I. Pakaya and M. Lestandy, "Comparison of Machine Learning Classification Methods in Hepatitis C Virus," *JOIN (Jurnal Online Informatika),* vol. 6, no. 1, pp. 73-78, 2021.