International Journal of Informatics, Data, and Emerging Applications
Vol 1, No 1, January – June, 2026, pp. 13-18
ISSN: xxxx-xxxx (Print) | E-ISSN: xxxx-xxxx

Corn Plant Disease Detection Using Deep Learning

Priscilla Elva Ruth Momongan
Master Program of Informatics Postgraduate Program
Sam Ratulangi University
Manado, Indonesia
priscillamomongan111@student.unsrat.ac.id

Nikita Veren Mamonto
Master Program of Informatics Postgraduate Program
Sam Ratulangi University
Manado, Indonesia
nikitamamonto111@student.unsrat.ac.id

Alya Melinda Johanis
Master Program of Informatics Postgraduate Program
Sam Ratulangi University
Manado, Indonesia
alyajohanis111@student.unsrat.ac.id

***Abstract*—Detection of diseases on corn leaves based on images requires a Convolutional Neural Network (CNN) capable of accurately recognizing visual patterns, as symptoms often appear similar across different disease classes. However, various CNN architectures need to be evaluated to determine which model performs optimally for corn leaf disease detection. In this study, three CNN architectures, namely VGG16, InceptionV3, and DenseNet, are compared to identify the most effective model for addressing the challenges of corn leaf disease classification. In addition, a transfer learning approach is employed to automatically extract relevant visual features. The models are evaluated based on Accuracy, Precision, Recall, and F1-score to determine the best-performing architecture. This comparison is important because each model differs in terms of layer depth, feature extraction strategy, and computational complexity, which can influence overall performance on the corn leaf dataset. The training process utilizes the Adam optimizer with a learning rate of 0.0001. The results indicate that VGG16 achieves training and validation accuracies of 90% and 91%, respectively, InceptionV3 achieves 92.10% and 93.02%, while DenseNet delivers the highest performance with 95.41% and 97.05%, respectively. DenseNet achieves superior performance due to its dense connectivity, which improves feature reuse and gradient flow. Therefore, DenseNet is considered the most effective model and has strong potential as the basis for developing an image-based automatic detection system for corn leaf diseases.**

Keywords—Convolutional Neural Network, Corn Leaves, Detection, Image Classification, Transfer Learning

I. INTRODUCTION

One of the most significant agricultural products that strategically supports both the national economy and food security is corn [1]. In Indonesia, corn is ranked among the top three staple crops alongside rice and soybeans and holds high economic value due to its steadily increasing demand each year [2]. Therefore, improving the productivity and quality of corn yields has become one of the primary focuses in the agricultural sector. However, corn productivity is often hindered by various leaf diseases caused by fungi and bacteria [3]. These diseases can reduce crop yields by significant percentages, especially when not detected at an early stage. Leaf diseases not only affect kernel quality but also disrupt the photosynthesis process, leading to overall stunted plant growth [4]. Delayed disease control efforts frequently result in substantial economic losses for both farmers and the agricultural industry.

Early detection of corn leaf diseases is a crucial step in preventing disease spread and minimizing yield losses. However, traditional methods that rely on visual observation by farmers or agricultural experts still face several limitations. Manual identification is time consuming, requires a high level of precision, and heavily depends on individual experience, making it prone to subjective errors and inconsistent results. On a large scale, these methods are also inefficient, as they are difficult to apply in real time across extensive agricultural fields [5], [6].

Currently, Convolutional Neural Networks (CNNs) demonstrate outstanding performance in various visual recognition tasks, including plant disease classification. CNNs are more adaptable to intricate image patterns that are challenging to recognize using traditional methods because they can automatically learn visual feature representations without the need for manual feature engineering [7]. However, the great visual similarities across several disease classes, such as overlapping color patterns, textures, and lesion shapes, makes it difficult to identify diseases on corn leaves. In order to extract discriminative features from maize leaf images, this situation requires the employment of models with high generalization capacity and efficient network topologies [8], [9].

The main contributions of this paper are as follows. First, this study provides a comprehensive experimental evaluation of three transfer learning–based CNN architectures, namely VGG16, InceptionV3, and DenseNet, for corn leaf disease detection . Second, all models are trained and validated using the same dataset and experimental settings to ensure a fair and objective performance comparison. Finally, the study identifies the most effective architecture based on multiple evaluation metrics, providing insights for the development of accurate and reliable plant disease detection systems. This comparison is essential because each architecture differs in layer depth, feature extraction strategy, and network complexity, which directly influence the model's capability to learn visual patterns from corn leaf datasets [10], [11]. In order to classify corn leaf diseases, this work provides a thorough comparison of three CNN-based transfer learning architectures: VGG16, INCEPTION-V3, and DenseNet. Using performance criteria such accuracy, precision, and recall, the examination investigates how parameter size and feature extraction techniques impact each model's capacity to identify visually similar illness patterns.

II. METHODOLOGY

This study follows a series of essential steps aimed at understanding and implementing an effective solution, starting with data collection to gather relevant information,

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

data preprocessing to clean and prepare the data, modeling to develop an appropriate model, and model evaluation to test and validate the model's performance.

A. Data Collecting

The dataset used in this study was obtained from the Kaggle website, titled Plant Village Dataset. As shown in Table I, a total of 4,360 images of corn leaf diseases were collected. Corn Common Rust, Corn Grey Spot, Northern Leaf Blight, and Healthy are the four categories into which this study divides the photos. The information is shown in Table I.

B. Data Pre-processing

Training, validation, and testing data are the three groups into which the dataset is separated. In total, the dataset consists of 4,360 images, comprising 3,200 for training, 800 for validation, and 360 for testing. The next step involves data preprocessing through data augmentation. In this stage, several augmentation techniques are applied, including rescaling, zooming, shifting, shearing, horizontal flipping, and filling. These methods are used to reduce the possibility of overfitting and to broaden the variety of training data. Furthermore, the augmentation process is carefully designed to preserve the original visual characteristics of the leaves. The details of this process are summarized in Table II.

C. CNN Proposed Model

The model used in this study employs transfer learning based on convolutional neural networks (CNNs). Convolution operations are used by the convolutional layer, the main part of CNN, to extract local features like edges from the input image. A tiny spatially limited subset of neurons is connected to each node in a convolutional layer. CNNs are a kind of neural network that can identify and detect objects in an image. They are frequently employed for picture data.

In this study, three types of CNN models are implemented using different CNN architectures to detect diseases in corn plants. The architectures used are VGG-16, INCEPTION-V3, and DenseNet. The goal is to determine which CNN model is the most effective and accurate for plant disease detection. Each architecture has its own unique characteristics and strengths in image processing and analysis, allowing for a comparative evaluation of their performance.

TABLE I. NUMBER OF IMAGE DATA OF CORN LEAF DISEASES

Type of Disease	***Quantity***
Common Rust	1090
Grey-Spot	1090
Northem Leaf Blight	1090
Healthy	1090
Quantity	4360

TABLE II. DATA AUGMENTATION CONFIGURATION

Category	***Value***
Rescale	1.0/255.0
Zoom Range	0.2
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.2
Horizontal Flip	0.2
Fill Mode	Nearest

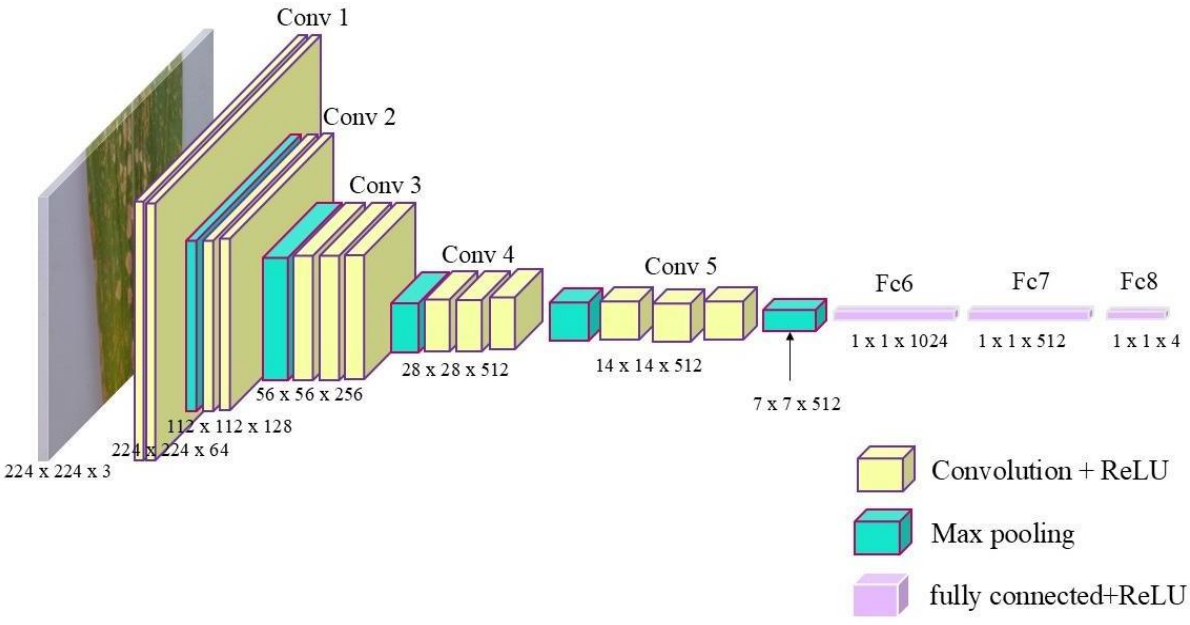

Fig. 1. Architecture of VGG. It employs a 3 × 3 convolutional layer to extract the information.

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

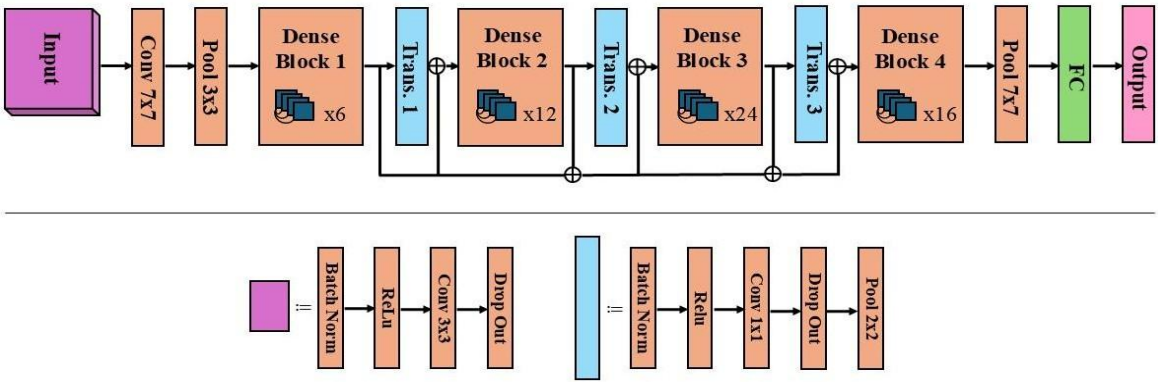

Fig. 2. Architecture of DenseNet Convolutional Neural Network.

TABLE III. DATA AUGMENTATION CONFIGURATION

Type	*Kernel Size / Stride*	*Input Size*
Convolution	3 x 3 / 2	224 x 224 x 3
Convolution	3 x 3 / 1	111 x 111 x 32
Convolution	3 x 3 / 1	109 x 109 x 32
Pooling (Max)	3 x 3 / 2	109 x 109 x 64
Convolution	3 x 3 / 1	54 x 54 x 64
Convolution	3 x 3 / 1	52 x 52 x 80
Convolution	3 x 3 / 1	52 x 52 x 192
Pooling (Max)	3 x 3 / 2	52 x 52 x 192
Inception Module	3 modules	25 x 25 x 256
Inception Module	5 modules	12 x 12 x 288
Inception Module	3 modules	5 x 5 x 2048
Pooling (Average)	Global Avarage Pooling	5 x 5 x 2048
Fully Connected (Dense)	1024 units	1 x 1 x 2048
Fully Connected (Dense)	512 units	1 x 1 x 1024
Output (Dense + Softmax)	4 units (classes)	1 x 1 x 512

1) VGG16

In this investigation, an RGB image measuring 224 × 224 × 3 is fed into the VGG-16 architecture. As seen in Fig. 1, the input is first processed in Conv 1, which creates a 224 × 224 × 64 feature map using two convolutional layers with 64 filters and a kernel size of 3 × 3. A 2 × 2 MaxPooling operation is then applied to reduce the spatial resolution to 112 × 112 × 64, enabling the model to concentrate on more pertinent data [12]. The process continues in Conv 2, which contains two convolutional layers with 128 filters to extract more complex visual features, resulting in an output of 112 × 112 × 128. Another MaxPooling layer reduces the dimension to 56 × 56 × 128. Conv 3 further enriches the feature representation with three convolutional layers containing 256 filters, producing 56 × 56 × 256 feature maps before spatial resolution is reduced again to 28 × 28 × 256 via MaxPooling.

In Conv 4, three convolutional layers with 512 filters learn higher level feature representations, yielding output dimensions of 28 × 28 × 512, which are then reduced to 14 × 14 × 512 using MaxPooling. Conv 5 follows a similar structure with three convolutional layers, each containing 512 filters, maintaining the resolution at 14 × 14 × 512. The final MaxPooling layer reduces this to 7 × 7 × 512, representing high level feature extraction that captures the global characteristics of the objects. The 7 × 7 × 512 feature maps are then aggregated using Global Average Pooling, producing a 1 × 1 × 512 feature vector. This vector is fed into the classification stage consisting of two fully connected layers with output dimensions of 1 × 1 × 1024 and 1 × 1 × 512, respectively. The final fully connected layer contains four units and uses a softmax activation function to generate the class prediction outputs [13], [14]. The consistent use of small convolutional kernels helps preserve spatial information while maintaining computational efficiency.

2) DenseNet

As illustrated in Fig. 2, the DenseNet architecture begins with an input image of size 224 × 224 × 3. This image is then processed through an initial 7 × 7 convolution with a stride of 2 to extract early features and reduce the spatial dimensions to 112 × 112. This is followed by a 3 × 3 MaxPooling with stride 2, further reducing the resolution to 56 × 56 before entering the core of the network [15]. After then, the network moves through four thick blocks in a row, each with six, twelve, twenty-four, and sixteen layers. Batch normalization, ReLU activation, a 1 × 1 convolution for channel efficiency, and a 3 × 3 convolution to produce new features make up each dense layer. The concatenation of each layer's output with all prior outputs inside the same block is a crucial feature of DenseNet that improves information flow and lessens the vanishing gradient issue. Transition layers made up of batch normalization, 1 × 1 convolution, and 2 × 2 average pooling minimize feature dimensions and preserve parameter efficiency in between dense blocks. Through these stages, the spatial dimensions gradually decrease from 56 × 56 to 28 × 28, 14 × 14, and finally 7 × 7, while the number of channels increases significantly. Global Average Pooling is used to aggregate the 7 × 7 × 1024 feature maps following the last dense block, resulting in a feature vector with one value per channel. Class probabilities are then produced by passing this vector through a fully connected layer and a softmax activation. DenseNet leverages dense connectivity, bottleneck convolutions, and transition layers to achieve high

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

parameter efficiency while maintaining substantial network depth, resulting in strong and stable performance for image classification tasks [16], [17].

3) INCEPTION-V3

In this study, the INCEPTION-V3 architecture begins with a 224 × 224 × 3 input image that is processed through multiple initial convolutional layers. The first convolution uses a 3 × 3 kernel with a stride of 2, reducing the image size to 111 × 111 × 32. This is followed by two additional 3 × 3 convolutions to enrich feature representations, resulting in 109 × 109 × 32. A 3 × 3 MaxPooling with stride 2 is then applied reducing the spatial dimensions to 109 × 109 × 64, marking the basic feature extraction stage [18]. The next stage consists of a series of advanced convolutions with 3 × 3 kernels, producing feature maps of 54 × 54 × 64, 52 × 52 × 80, and 52 × 52 × 192. These layers further deepen the feature mapping before entering the core of the architecture: the Inception Modules. The network moves through three stages of Inception modules after the second MaxPooling, which reduces the feature size to 25 × 25 × 256: 3 modules at 25 × 25 resolution, 5 modules at 12 × 12, and 3 modules at 5 × 5. INCEPTION-V3 Modules are the main building blocks that perform multiple convolutions and pooling operations in parallel within a single stage, including 1 × 1 convolutions for channel dimensionality reduction, 3 × 3 and 5 × 5 convolutions to capture multi scale patterns, and pooling branches that preserve global information. By concatenating the outputs from these concurrent branches, the model may learn both local and global features at the same time without needlessly adding more parameters. After all modules, the final feature map is 5 × 5 × 2048 and is processed using Global Average Pooling to summarize the information into a 1 × 1 × 2048 feature vector. This vector is then sent into two completely linked layers with 1024 and 512 units, respectively, to integrate high level information before classification, as Table III illustrates. The output layer is a Dense layer with four units and a Softmax activation based on the number of classes in the classification job adjusted for this study [19].

D. Model Training

Using the Adam optimizer as one of the training parameters, the model was trained for 25 epochs with a batch size of 80 and a learning rate of 0.0001. The loss function employed in this work was categorical cross-entropy because it deals with a multi-class classification problem. The output layer used the Softmax activation function to provide probability values for each class in the four-class classification task. The model was trained and evaluated using the Python TensorFlow framework [20].

E. Model Evaluation

Several metrics were used to evaluate the model. Accuracy shows overall correctness. Precision shows how many predicted positives are correct, while recall shows how many actual positives are identified. The F1-score combines precision and recall for a balanced evaluation.

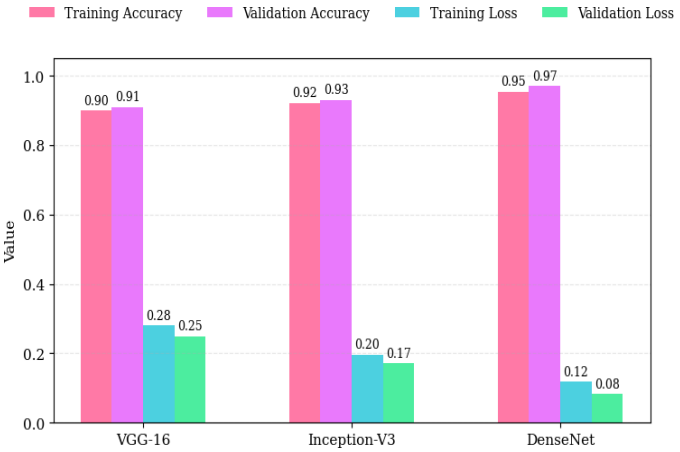

Fig. 3. Performance of CNN Models Based on Accuracy and Loss Values During Training and Validation.

TABLE IV. COMPARISON OF CLASSIFICATION REPORT METRICS ACROSS CNN ARCHITECTURE

Architecture	*Accuracy*	*Precision*	*Recall*	*F1-Score*
VGG-16	0.9222	0.93	0.92	0.92
INCEPTION-V3	0.9305	0.93	0.93	0.93
DenseNet	**0.9750**	**0.98**	**0.97**	**0.97**

III. RESULTS AND DISCUSSIONS

A. Model Performance Analysis

Fig. 3 compares three CNN architectures: DenseNet, INCEPTION-V3, and VGG-16. The four primary performance metrics used to evaluate these models are training accuracy, validation accuracy, training loss, and validation loss. The figure highlights each model's learning capacity on the training dataset as well as its generalization performance on unseen validation data. Among the three architectures, DenseNet demonstrates the most outstanding performance, achieving training and validation accuracy values that are nearly perfect, approaching 100 percent, along with the lowest training and validation loss values.

VGG-16 shows the weakest overall performance among the three architectures. Lower accuracy and higher loss values indicate limited capability in modeling complex data patterns, which can be attributed to its simpler architecture and reduced interlayer connectivity. These structural limitations restrict information propagation and gradient flow during training, resulting in slower convergence and reduced generalization performance.

INCEPTION-V3 performs better than VGG-16 but remains slightly below DenseNet, with relatively low loss values. This indicates that INCEPTION-V3 effectively captures deep hierarchical features through its Inception Modules, although its optimization efficiency remains comparatively lower. In summary, DenseNet demonstrates the best overall performance among the evaluated models.
Its densely connected structure facilitates efficient information flow across layers, promotes deep feature learning, and enhances model stability and accuracy during both training and validation phases.

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

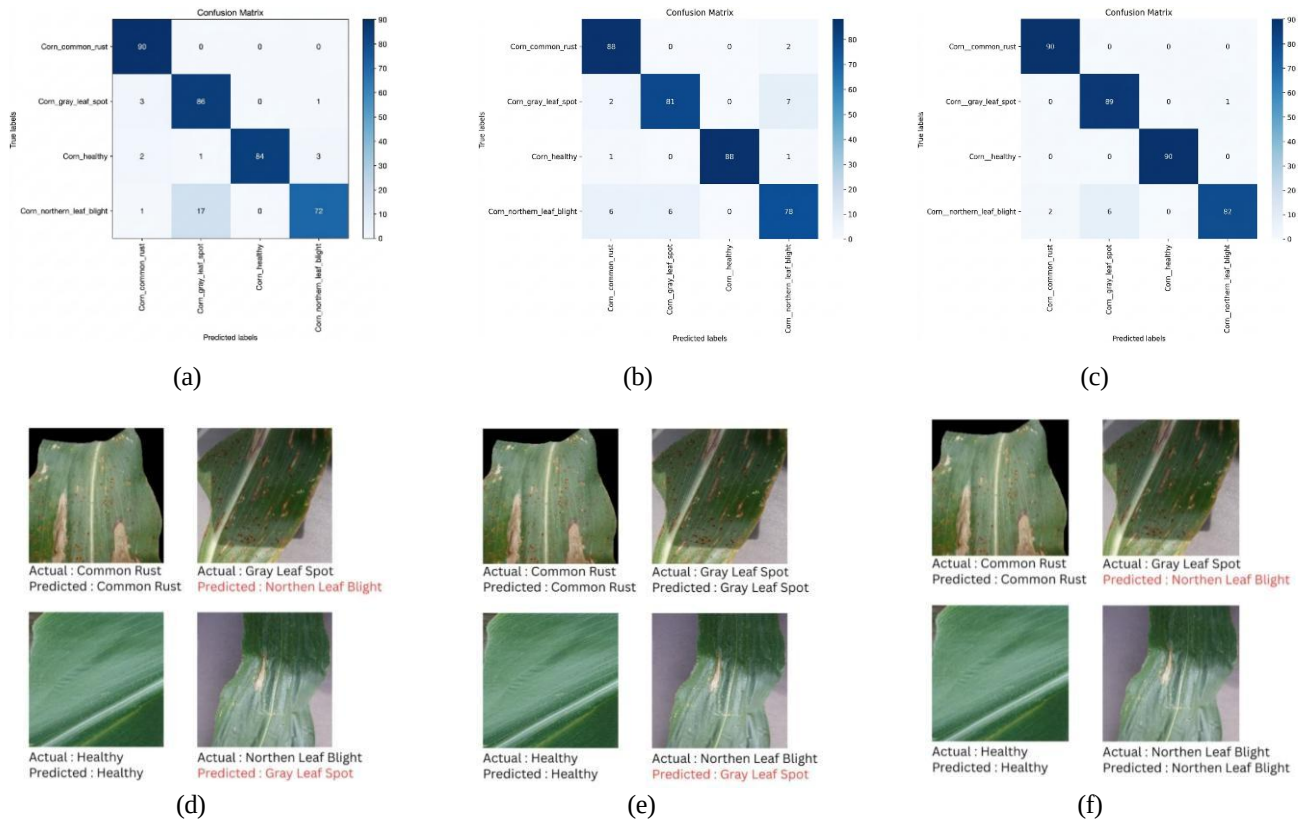

Fig. 4. Evaluation results of corn leaf disease classification using three CNN architectures. (a) Confusion matrix of VGG-16. (b) Confusion matrix of INCEPTION-V3. (c) Confusion matrix of DenseNet. (d) Qualitative examples of correct and incorrect predictions of VGG-16. (e) Qualitative examples of correct and incorrect predictions of INCEPTION-V3. (f) Qualitative examples of correct and incorrect predictions of DenseNet.

B. Classification Report Analysis

Table IV displays the classification report findings for three convolutional neural network architectures: VGG-16, InceptionV3, and DenseNet. Four key performance criteria were used to evaluate these architectures: F1-Score, Accuracy, Precision, and Recall. These metrics provide a comprehensive assessment of each model's ability to perform accurate and trustworthy classification. Overall, all models perform well, indicating reliable categorization skills.

The classification report metrics for each of the three CNN designs are compared in Table IV. With an accuracy of 0.9750, precision of 0.98, recall of 0.97, and F1-score of 0.97, DenseNet consistently ranks first on all criteria. These results indicate that DenseNet attains an optimal balance between correctly identifying positive samples and minimizing false predictions, resulting in stable and efficient classification performance. Inception-V3 is ranked second with an accuracy of 0.9305 and balanced precision, recall, and F1-score values of 0.93. VGG-16 performs somewhat worse, with an accuracy of 0.9222 and precision, recall, and F1-score values of 0.93, 0.92, and 0.92, respectively. This comparison shows that while both models are capable of efficiently classifying data, DenseNet still outperforms them in terms of maximizing the trade-off between precision and recall. Overall, Table IV's results support the notion that DenseNet is the best architecture out of the three models. Its dense connectivity structure enables more efficient information and gradient flow across layers, enhancing the model's ability to learn deeper and more discriminative feature representations, ultimately leading to superior classification performance.

C. Confusion Matrix

Based on the model evaluation results in the confusion matrices in Fig. 4, the performance of the three CNN architectures in classifying the four corn leaf types can be compared. The confusion matrix in Fig. 4(a) shows that the VGG-16 architecture performs well across all classes. It correctly identified 90 samples of Corn Common Rust with no errors, 86 of Gray Leaf Spot with 4 misclassified, 84 of Healthy leaves, and 72 of Northern Leaf Blight, which was often confused with Gray Leaf Spot due to similar lesion patterns. Examples of accurate and inaccurate predictions are shown in Fig. 4(d). Common Rust and healthy leaves were accurately detected, however Gray Leaf Spot and Northern Leaf Blight were the most frequently misclassified. Overall, VGG-16 performs well but still needs improvement through data augmentation and a more diverse dataset to better distinguish visually similar diseases.

The INCEPTION-V3 architecture provides robust and balanced classification performance, as shown by the confusion matrix in Fig. 4(b). Eighty-eight samples of Common Rust with just two errors, eighty-one samples of Gray Leaf Spot with slight misclassifications into Common Rust and Northern Leaf Blight, and eighty-eight samples of Healthy with no errors were all properly recognized by the model. Northern Leaf Blight had 78 correct predictions, although some samples were misclassified into the other two

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/

categories with similar symptoms. As a visual representation of the model's predictions, Fig. 4(e) demonstrates that INCEPTION-V3 can accurately detect Common Rust and Healthy, but it struggles to differentiate between Gray Leaf Spot and Northern Leaf Blight. Hence, although Inception-V3 shows strong performance and good generalization, further optimization through fine tuning and more diverse data augmentation is recommended to improve accuracy for classes with overlapping visual characteristics.

The confusion matrix in Fig. 4(c) reveals that DenseNet achieved the best performance among the three models. It correctly classified all 90 samples of Common Rust and Healthy without any errors, 89 samples of Gray Leaf Spot with only one misclassification into Northern Leaf Blight, and 82 samples of Northern Leaf Blight with minimal errors into the other two categories. As a visual representation of the model's performance, Fig. 4(f) shows that DenseNet accurately identified most images, with only minor errors occurring in cases where pattern similarities existed. This advantage results from the dense connectivity mechanism, which facilitates feature reuse and efficient gradient flow across layers. Overall, this model demonstrates the best generalization capability, though expanding the dataset diversity could further enhance its accuracy in distinguishing diseases with similar visual symptoms.

IV. Conclusion

This study compared three convolutional neural network architectures, namely VGG16, INCEPTION-V3, and DenseNet, for detecting corn leaf diseases using a transfer learning approach. Experimental results on the PlantVillage dataset, consisting of 4,360 images, show that the DenseNet model achieved the best performance, reaching a training accuracy of 95% and a classification accuracy of 97.50%, with precision, recall, and F1-score of 0.98, 0.97, and 0.97, respectively. The superiority of DenseNet is attributed to its dense connection mechanism, which enables each layer to receive information from all preceding layers. This structure enhances gradient flow efficiency, maximizes feature utilization, and reduces the risk of losing important information during the feature processing stage. Therefore, DenseNet has strong potential to serve as a reliable foundation for developing automatic image-based plant disease detection systems in agriculture. Future research is recommended to extend model evaluation to larger and more diverse datasets, increase the number of disease classes, and explore the application of CNN based approaches to other plant disease detection tasks to assess their adaptability and effectiveness.

References

[1] Q. N. Azizah, "Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet," *sudo J. Tek. Inform.*, vol. 2, no. 1, pp. 28–33, 2023, doi: 10.56211/sudo.v2i1.227.

[2] Mutia Fadhilla, D. Suryani, A. Labellapansa, and H. Gunawan, "Corn Leaf Diseases Recognition Based on Convolutional Neural Network," *IT J. Res. Dev.*, vol. 8, no. 1, pp. 14–21, 2023, doi: 10.25299/itjrd.2023.13904.

[3] Y. A. Sitorus, Y. A. Sari, and S. Adinugroho, "Evaluasi Komparatif Arsitektur Lightweight CNN, MobileNetV2, dan EfficientNetB0 dalam Deteksi Penyakit Daun Jagung," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 8, pp. 1–6, 2025, [Online]. Available: http://j-ptiik.ub.ac.id.

[4] P. F. Johari, N. Arifin, M. Muzaki, and M. S. A. Utama, "Corn Leaf Diseases Classification Using CNN with GLCM, HSV, and L*a*b* Features," *J. Tek. Inform.*, vol. 6, no. 2, pp. 709–722, 2025, doi: 10.52436/1.jutif.2025.6.2.4345.

[5] G. Gumelar *et al.*, "Jurnal Computer Science and Information Technology (CoSciTech) Implentation of CNN for Corn Leaf Disease Identification," vol. 6, no. 2, pp. 175–180, 2025.

[6] G. A. Sandag, F. M. Tombeng, and S. R. Vridvly, "CornNet : Implementation of Transfer Learning for Disease Classification in Corn Plants Based on Web Application," vol. 11, no. 1, pp. 167–181, 2025.

[7] A. Zidha and W. Indah, "Perbandingan Performa Algoritma VGG16 dan VGG19 Melalui Metode CNN Untuk Klasifikasi Varietas Beras," vol. 4, no. 2, 2023.

[8] D. Pratama, "Jurnal Teknologi Terpadu LEARNING," *J. Teknol. Terpadu*, vol. 8, no. 1, pp. 89–94, 2022.

[9] J. R. Aisya and A. Prasetiadi, "Klasifikasi Penyakit Daun Kentang dengan Metode CNN dan RNN," *J. Tekno Insentif*, vol. 17, no. 1, pp. 1–10, 2023, doi: 10.36787/jti.v17i1.888.

[10] M. T. Ahad, Y. Li, B. Song, and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artif. Intell. Agric.*, vol. 9, pp. 22–35, 2023, doi: 10.1016/j.aiia.2023.07.001.

[11] L. Kong and J. Cheng, "Classification and detection of COVID-19 X-Ray images based on DenseNet and VGG16 feature fusion," *Biomed. Signal Process. Control*, vol. 77, no. January, p. 103772, 2022, doi: 10.1016/j.bspc.2022.103772.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[13] S. Tammina, "Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images," *Int. J. Sci. Res. Publ.*, vol. 9, no. 10, p. p9420, 2019, doi: 10.29322/ijsrp.9.10.2019.p9420.

[14] L. Yang, S. Xu, X. Y. Yu, H. Bin Long, H. H. Zhang, and Y. W. Zhu, "A new model based on improved VGG16 for corn weed identification," *Front. Plant Sci.*, vol. 14, no. July, pp. 1–11, 2023, doi: 10.3389/fpls.2023.1205151.

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 2261–2269, 2017, doi: 10.1109/CVPR.2017.243.

[16] A. Haque, I. Sutradhar, M. Rahman, M. Hasan, and M. Sarker, "Convolutional Nets for Diabetic Retinopathy Screening in Bangladeshi Patients," no. August, 2021, [Online]. Available: http://arxiv.org/abs/2108.04358.

[17] T. Zhou, X. Ye, H. Lu, X. Zheng, S. Qiu, and Y. Liu, "Dense Convolutional Network and Its Application in Medical Image Analysis," *Biomed Res. Int.*, vol. 2022, 2022, doi: 10.1155/2022/2384830.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.

[19] J. Thomkaew and S. Intakosum, "Improvement Classification Approach in Tomato Leaf Disease using Modified Visual Geometry Group (VGG)-InceptionV3," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, pp. 362–370, 2022, doi: 10.14569/IJACSA.2022.0131244.

[20] T Andriani and M Iqbal, "Penggunaan Transfer Learning untuk Peningkatan Akurasi Deteksi Penyakit Tanaman," *J. Multimed. dan Teknol. Inf.*, vol. 07, no. 02, pp. 90–95, 2025.

Received: 26 February 2026, Revision: 27 Maret 2026, Accepted: 10 April 2026
available at: https://ejournal.unsrat.ac.id/v3/index.php/idea/