A Comparison of Light CNN-29 and SqueezeNet for Leaf Disease Classification

Yulianto Yulianto1, a), Ida Bagus Ananta wijaya2, b), Bambang Kartono Kurniawan2, c), Gabriella Sagita Putri3, d) and Nor Hamizah Abdul Hamid4, e)

Author Affiliations

1Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia.

2Interior Design Department, School of Design, Bina Nusantara University, Jakarta 11480, Indonesia.

3Public Relations Department, Faculty of Digital Communication and Hotel&Tourism, Bina Nusantara University, Jakarta 11480, Indonesia.

4Sustainability, Urban Design and Wellbeing research group, Faculty of Architecture and Ekistics, Universiti Malaysia Kelantan, Bachok 16310, Malaysia.

Author Emails

a)Corresponding author: [yulianto003@binus.ac.id](mailto:yulianto003@binus.ac.id)   
b)[ida.wijaya@binus.ac.id](mailto:ida.wijaya@binus.ac.id)

c)[bambang.k@binus.ac.id](mailto:bambang.k@binus.ac.id%20)

d)[gabriella.putri004@binus.ac.id](mailto:gabriella.putri004@binus.ac.id%20)

e)[norhamizah@umk.edu.my](mailto:norhamizah@umk.edu.my)

**Abstract.** The process of manually identifying leaf diseases has been automated using artificial intelligence, which promises to speed up the identification process and increase its accuracy. Accurate classification of leaf diseases is essential for determining the most effective methods of preventing their spread to entire crops, which can reduce yield quality and quantity. This study evaluates and compares the performance of two deep learning architectures, Light CNN-29 and SqueezeNet, for classifying leaf diseases. The study addresses the critical need for accurate, automated disease identification to promote sustainable agriculture and ensure food security, particularly by preventing significant decreases in yield. This effort aligns with Sustainable Development Goal (SDG) number two. While many deep learning models have achieved high accuracy in this area, this study examines the effectiveness of Light CNN-29. Light CNN-29, originally designed for face image recognition, has not yet been applied to leaf disease classification. The sults showed that the two models had different abilities to generalize. Light CNN-29 achieved 100% accuracy on the training data yet performed substantially worse on the validation and test sets achieving only 93% accuracy. This discrepancy suggests that the model overfit the training data. In contrast, SqueezeNet demonstrated superior performance, achieving 99% accuracy on the training dataset and 97% on the validation and test dataset. It also maintained high precision, recall, and F1 scores of 97% on unseen test dataset. In conclusion, Light CNN-29 shows potential, achieving an accuracy rate of 93%.

**Keywords:** SDG Number 2, Sustainable Agriculture, Leaf disease classification, Light CNN-29, Image Classification.

# INTRODUCTION

The goals of Sustainable Development Goal (SDG) number 2 are achieving food security and promoting sustainable agriculture [1]. One of agriculture’s fundamental roles is producing food, which is an important sector to protect because it is a basic human need. Agricultural productivity must increase to meet the needs of a growing global population. SDG number 2 encourages society to cultivate a variety of fruits, vegetables, and other crops to combat micronutrient and macronutrient deficiencies which can have severe and long-lasting health consequences. The productivity of agriculture also depends on crops that can withstand threats such as pests, bacteria, and fungi that can impact plant growth [2]. Threats have the potential to impact crop yield, reducing the quality or quantity of produce [3]. One type of threat is disease, which can lead to a decrease in yield production [4].

The initial identification of threats to plants can be made by examining the condition of their leaves. As a note, although the visual indication of the disease can be observed on the leaves, the disease can also affect the stems, fruits, and even the roots [5]. The goal of identifying leaf diseases is to prevent them from spreading to the entire crop because the first mitigation can be done immediately [6]. The observation of leaf conditions sometimes requires the presence of an expert, which automatically incurs extra costs [6].

Recent research has proposed several deep learning architectures to overcome the manual identification of leaf diseases, achieving satisfactory accuracy scores. For example, ECA-ResNet34 scored 98,5% accuracy [2], Handcraft CNN scored 98.18% accuracy [3], VGG16 scored 97% accuracy [5], ShuffleNet V2 with accuracy 99,43% [4], improved AlexNet with score 93.6% [7], ResNet101 with attention network give score 99.82% [8], Tiny\_Exception with score 94,3% [9], and Vision Transformer scored the highest accuracy of 99.77% [10].

Although deep learning architectures have recently achieved satisfactory accuracy, many AI models remain unexplored, such as those used for face image classification. This study aims to compare the performance of two deep learning architectures, particularly in terms of classification accuracy. The study will compare the Light CNN-29 architecture with the Squeeze Net architecture for feature extraction, combined with a fully connected neural network layer for classification. This study chose the Light CNN-29 because this model has never been used for leaf disease image classification. Besides that, this study reuses the SqueezeNet architecture for comparison because its model was previously proposed with 99.92% accuracy and a fast-training process for leaf disease classification [3].

# LITERATURE REVIEW

Kathole et al. [11] state in their research that AI based on deep learning requires a large amount of data to enable the model architecture to provide classification results with minimal misclassification. However, gathering and annotating data is often time-consuming and expensive. Based on this problem, the authors propose a few-shot learning model to handle small training datasets. Another case inspiring the proposal of few-shot learning due to the limited leaf disease samples is the Swin-Transformer V2-F6 architecture. The results show that the accuracy can reach 91.81% [12].

The research on leaf disease classification, proposed by Ouamane et al. [10], discusses the use of a Vision Transformer for plant disease classification to overcome the risk of human error and inaccurate identification due to observer bias in the manual process. The results showed achieve an accuracy of 99.77% in leaf disease classification. Inspired by the important case of identifying tea leaf disease to ensure proper treatment, Liang et al. [13] proposed EfficientNet Lite0 for leaf disease classification. The classification accuracy achieved was 94.34%. The use of Dy-Tri ResNet50 for disease classification has been proposed as a way to prevent decreases in maize yield caused by disease. This approach has been shown to have an accuracy of 98.79% as reported in a study by Tang et al. [14].

The tomato plant is susceptible to several diseases caused by various factors, such as bacteria, pathogens, fungi, and other virus organism [15]. A combination of a feature extractor based on a hand-crafted HOG model and a classification using an LBP algorithm was proposed to recognize disease on tomato leaves with varying characteristics. The proposed method achieved an accuracy of 98.72% [16]. Another research combination proposes a three-model approach: segmentation using a U-Net combined with VGG-16 as a feature extractor and a T-LSTM with an attention mechanism for classification. This approach mechanism for classification. This approach can achieve an accuracy of 99.98% [15].

Training the AI model for leaf classification requires special attention to the dataset because threats such as adversarial leaf images need to be filtered to prevent bias or an increased false classification rate in the AI model. CL-CondensNetV2 which combines CA attention and CondenseNetV2 was developed to address this issue. Using the Plant Village dataset, classification accuracy reached 99% [17].

# METHOD

The Light CNN-29 architecture is famous for its use in classification related to biometric attributes. The architecture has been proposed for use in face image classification for the first time [18]. The Light CNN-29 uses a Max Feature Map (MFM) instead of a conventional activation function [18]. The MFM layer operates by convolving the input layer twice, which allows it to split the output layer into two groups [18]. Equation (1) is the MFM function. In which the feature output layer ) is derived from the splitting process following convolution into two groups of layers, each comprising 50% of the total. The first group of layers is denoted and the second group is denoted . The max function only passes the group of layers with the highest output [18]. This ensures that the MFM will always pass information and never block it in forward operation [18]. FIGURE 1 shows that the Light CNN-29 utilizes an MFM layer in Layers 1, 2, 3, 5, 6, 8, 6, 8, 9, 10, and 11. The architecture also utilizes the residual block, which consists of a combination of two MFM blocks connected in series and equipped with a skip connection [18]. Residual Block MFM is positioned on layers 2, 5, 8 and 10. The Light CNN-29 resolution input image in FIGURE 1 is designed to receive an incoming image with a resolution of 320 x 426 pixels. The resolution input will affect the output feature after Layer 12 and the flattening process resulting in 271.360 output features. These features are then fed to a fully connected layer feature. These features are then fed to a fully connected layer with 1.024 neurons, resulting in a 15-class output. The Light CNN-29 detail parameter can be seen in TABLE 1.



Layer 2

Residual Block MFM

Layer 3

MFM Group

Layer 4

Max Pooling + Average Pooling

Layer 5

Residual Block MFM

Layer 6

MFM Group

Layer 7

Max Pooling + Average Pooling

Layer 8

Residual Block MFM

Layer 9

MFM Group

Layer 10

Residual Block MFM

Layer 11

MFM Group

Layer 12

Max Pooling + Average Pooling

Flattened

Layer 1 MFM

Class 1

Class 2

Class n

FIGURE 1. The Light CNN-29 architecture is used as a feature extractor in combination with a fully connected layer as a classifier.

TABLE 1. Details of the Light CNN-29 parameter from FIGURE 1.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Layer Position | Input Channels | Output Channels | Kernel Size | Stride | Padding | Number of Blocks |
| Layer 1 MFM | 3 | 48 | 3 x 3 | 1 | 1 | - |
| Layer 2 Residual block MFM | 48 | 48 | 3 x 3 | 1 | 1 | 1 |
| Layer 3 MFM group | 48 | 96 | 3 x 3 | 1 | 1 | - |
| Layer 4 Max pooling layer + Average pooling layer | 96 | 96 | 2 x 2 | 2 | 0 | - |
| Layer 5 Residual block MFM | 96 | 96 | 3 x 3 | 1 | 1 | 2 |
| Layer 6 MFM group | 96 | 192 | 3 x 3 | 1 | 1 | - |
| Layer 7 Max pooling layer + Average pooling layer | 192 | 192 | 2 x 2 | 2 | 0 | - |
| Layer 8 Residual block MFM | 192 | 192 | 3 x 3 | 1 | 1 | 3 |
| Layer 9 MFM group | 192 | 128 | 3 x 3 | 1 | 1 | - |
| Layer 10 Residual block MFM | 128 | 128 | 3 x 3 | 1 | 1 | 4 |
| Layer 11 MFM group | 128 | 128 | 3 x 3 | 3 | 1 | - |
| Layer 12 Max pooling layer + Average pooling layer | 128 | 128 | 2 x 2 | 2 | 0 | - |
| Flattened | - | 271360 | - | | | |
| Fully Connected Neural Network | 271360 | 1024 | - | | | |
| Fully Connected Neural Network | 1024 | 15 | - | | | |

|  |  |
| --- | --- |
|  | (1) |

The second AI architecture model used for comparison in this study is SqueezeNet. In a previous experiment, it achieved an accuracy of 99.92% [3]. A slight modification was also attempted by setting the input resolution to 320 x 426 pixels, which influenced the number of features output by SqueezeNet to 1000 feature vectors. The output was then fed into a fully connected neural network layer with 15 outputs representing the number of classes that match the number of classes in the Plant Village dataset.

This study used the Plant Village dataset [19]. The Plant Village dataset is available on their official website at plantvillage.psu.edu [7]. In leaf disease identification using computer vision, particularly image classification based on a deep learning architecture, the Plant Village dataset is one of the most famous datasets used for training AI model and reference in recent research [20][5][21][1]. This dataset contains square images with a resolution of 256 x 256 pixels, captured under highly controlled laboratory conditions. For example, each image contains only one leaf, which is positioned in the middle. The dataset contains three different leaf crops: leaf pepper, potato, and tomato. The total number of images in the dataset is 20.638. The number of leaf images for each subclass is not balanced. The detail class name and number of images for each class can be seen in TABLE 2.

TABLE 2. Details of the class name and the number of images provided by the Plant Village dataset for each class.

|  |  |
| --- | --- |
| Class name | Number of images |
| Tomato yellow leaf curl virus | 3208 |
| Tomato target spot | 1404 |
| Tomato spider mites two spotted | 1676 |
| Tomato septoria leaf spot | 1771 |
| Tomato mosaic virus | 373 |
| Tomato leaf mold | 952 |
| Tomato late blight | 1909 |
| Tomato healthy | 1591 |
| Tomato early blight | 1000 |
| Tomato bacterial spot | 2127 |
| Potato late blight | 1000 |
| Potato healthy | 152 |
| Potato early blight | 1000 |
| Pepper bell healthy | 1478 |
| Pepper bell bacterial spot | 997 |
| Total Images | 20638 |

# RESULT AND DISCUSSION

TABLE 3. Number of images for each class after the splitting process into the training 60%, validation 20%, and test sets 20% from the Plant Village dataset.

|  |  |  |  |
| --- | --- | --- | --- |
| Class name | Train set | Validation set | Test Set |
| Tomato yellow leaf curl virus | 1925 | 641 | 642 |
| Tomato target spot | 842 | 281 | 281 |
| Tomato spider mites two spotted | 1006 | 335 | 335 |
| Tomato septoria leaf spot | 1062 | 355 | 354 |
| Tomato mosaic virus | 224 | 75 | 74 |
| Tomato leaf mold | 571 | 190 | 191 |
| Tomato late blight | 1145 | 382 | 382 |
| Tomato healthy | 955 | 318 | 318 |
| Tomato early blight | 600 | 200 | 200 |
| Tomato bacterial spot | 1276 | 425 | 426 |
| Potato late blight | 600 | 200 | 200 |
| Potato healthy | 91 | 30 | 31 |
| Potato early blight | 600 | 200 | 200 |
| Pepper bell healthy | 887 | 296 | 295 |
| Pepper bell bacterial spot | 598 | 200 | 199 |
| Total Images | 12382 | 4128 | 4128 |

The pre-processed dataset used in this study was split into three parts: 60% for training, 20% for validation, and 20% for testing. The total proportion of each category after the splitting process can be observed in TABLE 3. After splitting data, all the leaf disease images with a resolution of 256 x 256 pixels were resized to a portrait with a width of 320 pixels and a height of 426 pixels. The aim of decreasing the image size was to reduce the computational load and slightly increase the speed, not only for training, validation, and testing. One method that can be applied to minimize the effect of overfitting on the AI model is to transform the image to normalize it before inserting it into the model. This can be done using the mean and standard deviation parameters [17]. The mean and standard deviation parameters in this study were gathered from the train data with 60% splitting proportion. The mean calculation resulted in a value of ([0.85573785, 0.83802794, 0.83231171]) and the standard deviation resulted in a value of ([0.24511901, 0.26017573, 0.26463725]). Those mean and standard deviation scalar values were used to normalize all the leaf images in the training, validation, and test sets of the PlantVillage dataset. The following parameters were applied to both tested models: train batch size of 16, number of epochs of 40, and an optimizer of stochastic gradient descent (SGD) utilized with a learning rate of 0.0001, momentum of 0.9, and weight decay of 0.0001. The loss function used for multi-class classification in this study was cross-entropy loss. More information about the hyperparameters used in this experiment can be found in TABLE 4. For more information, the Light CNN-29 model used in this study was handcrafted. For the SqueezeNet model, transfer learning was used with the pretrained parameter set to true.

TABLE 4. The hyperparameter was used to train both the Light CNN-29 model and the SqueezeNet model.

|  |  |
| --- | --- |
| Parameter Name | Parameter Value |
| Number of Epoch | 40 |
| Train Batch size | 16 |
| Optimizer | Stochastic Gradient Descent (SGD) |
| Learning Rate | 0.0001 |
| Momentum | 0.9 |
| Weight decay | 0.0001 |
| Loss function | Cross Entropy Loss |

A graph with lines and numbers

AI-generated content may be incorrect.

FIGURE 2. Train and validation accuracy of Light CNN-29 for each epoch from 1 to 40.

FIGURE 3 illustrates the model’s training and validation accuracy over 40 epochs. The model used a pre-trained SqueezeNet as a feature extractor followed by a fully connected layer as a classifier. The training and validation data were also split in the same way as in the first experiment using the Plant Village dataset. The blue line represents the training accuracy and the orange line represents the validation accuracy. On the first epoch, the training accuracy rose from approximately 81% to 99% by the 40th epoch. Similarly, the validation performance of classification began at around 82% and closely followed the trajectory of the training curve. At the final epoch (40), the training and validation accuracies were 98% and closely followed the trajectory of the training curve. At the final epoch (40), the training and validation results of Light CNN-29 displayed in FIGURE 2, SqueezeNet never reached 100% accuracy. However, the minimal and often negligible gap between the training and validation curves during the training process indicates that the model did not overfit.

A graph with blue and orange lines

AI-generated content may be incorrect.

FIGURE 3. shows the training and validation of SqueezeNet as a feature extractor combined with a fully connected layer as a classifier.

Following the training process, the performance of both the Light CNN-29 and SqueezeNet models was evaluated using the training dataset. FIGURE 4 shows the classification mapping into a confusion matrix. The Light CNN-29 model achieved 100% accuracy (FIGURE 4 (a)), while SqueezeNet performed well with minor errors (FIGURE 4 (b)). The Light CNN-29’s perfect score suggests that it may be overfitting by memorizing the training data. In contrast, SqueezeNet’s slight imperfections could indicate a more generalized model. Therefore, it is crucial to evaluate the models on validation and test datasets to determine their actual performance on new data.

FIGURE 5 shows the classification results for the Light CNN-29 and SqueezeNet models on the validation dataset. Light CNN-29’s performance (FIGURE 5 (a)) declined noticeably from its perfect training score, with precision, recall, and the F1 score dropping to 93%. SqueezeNet’s confusion matrix (FIGURE 5 (b)) also shows some misclassifications, but its overall performance metrics were higher than Light CNN-29’s (see

TABLE 6).

The ultimate evaluation of the two models in this study utilizes a test dataset comprising images that were not observed by either model during training or validation. FIGURE 6 presents the classification performance of the Light CNN-29 and SqueezeNet architectures in the form of confusion matrices. FIGURE 6 (a) shows the Light CNN-29 model’s confusion matrix. There is a noticeable spread of misclassifications in the off-diagonal cells. FIGURE 6 (b) shows that SqueezeNet performs much better. The vast majority of predictions are concentrated along the main diagonal and there are significantly fewer misclassifications in the off-diagonal cells that in the Light CNN-29 model. Specifically, only one category has the same correct score for both models, which can correctly classify 316 images of healthy leaf tomatoes.

|  |  |
| --- | --- |
|  |  |
| (a) Light CNN-29 | (b) SqueezeNet |

FIGURE 4. The classification mapping results are shown in the form of a confusion matrix using the training dataset. Figure (a) shows the results from the Light CNN-29 architecture ang Figure (B) shows the results from the SqueezeNet architecture.

|  |  |
| --- | --- |
|  |  |
| (a) Light CNN-29 | (b) SqueezeNet |

FIGURE 5. Shows the mapping results of the classification in the form of a confusion matrix using a valid dataset. Figure (a) shows the results from the Light CNN-29 architecture and figure (b) shows the results from the SqueezeNet architecture.

|  |  |
| --- | --- |
|  |  |
| (a) Light CNN-29 | (b) SqueezeNet |

FIGURE 6. Shows the mapping classification results of the classification in the form of a confusion matrix using the test dataset. Figure (a) shows the results from the Light CNN-29 architecture and Figure (b) shows the results from the SqueezeNet architecture.

After the 40-epoch training process was complete, we evaluated the performance of the proposed Light CNN-29 model and the SqueezeNet benchmark model in classifying plant leaf diseases using the Plant Village dataset. Performance was evaluated based on four key metrics-precision, recall, F1-score, and accuracy-across the training, validation, and test datasets. The quantitative comparison results of the experiments are summarized in TABLE 5,

TABLE 6, TABLE 7, and TABLE 8. On the training dataset (TABLE 5), the Light CNN-29 model achieved perfect scores of 100% for precision, recall, and the F1 score. SqueezeNet also performed well, achieving 99% on these same metrics. These results suggest that both models were able to learn features from the training leaf images. However, the perfect training score may be overfit. For the validation dataset (

TABLE 6), a distinction in performance emerged between the two models. SqueezeNet maintained high performance, achieving 98% precision, 97% recall, and an F1-score of 97%. In contrast, Light CNN-29’s performance dropped to 93% across all three metrics. The most critical evaluation, which was conducted using an unseen test dataset (see TABLE 7 and TABLE 8), confirmed these findings. SqueezeNet consistently outperformed the Light CNN-29 model. SqueezeNet achieved a test accuracy of 97%, as well as a precision, recall, and F1-score of 97% in comparison, the Light CNN-29 model recorded a test accuracy of 93%, with a precision, recall, and F1-score of 93%, 93%, and 92% respectively.

The limitations of this study were finally expressed as follows: the comparison between the two architectures was only tested using test data. While all the image data in the PlantVillage dataset contains images collected under highly controlled laboratory conditions. Testing in real cases with different specifications of cameras, lighting, and angels has not yet been discussed. The slight modifications to the Light CNN-29 model in this study were only to the input image resolution and the number of fully connected neurons. The data transformation imposed on the PlantVillage dataset was only to normalize imbalanced data.

TABLE 5. Weighted average of precision, recall, and F1 score for Light CNN-29 and SqueezeNet using training dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Precision | | Recall | | F1 Score | |
| Light CNN-29 | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** |
| 100% | 99% | 100% | 99% | 100% | 99% |

TABLE 6. Weighted average of precision, recall, and F1 score for Light CNN-29 and SqueezeNet using the validation dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Precision | | Recall | | F1 Score | |
| Light CNN-29 | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** |
| 93% | 98% | 93% | 97% | 93% | 97% |

TABLE 7. Weighted average of precision, recall, and F1 score for Light CNN-29 and SqueezeNet using test dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Precision | | Recall | | F1 Score | |
| Light CNN-29 | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** |
| 93 | 97% | 93% | 97% | 92% | 97% |

TABLE 8. Training, validation, and accuracy for Light CNN-29 and SqueezeNet.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Training Accuracy | | Validation Accuracy | | Test Accuracy | |
| Light CNN-29 | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** | **Light CNN-29** | **SqueezeNet** |
| 100% | 99% | 93% | 97% | 93% | 97% |

# CONCLUSION

In this study, the experimental implementation of the Light CNN-29, which has been used before for face image classification was tested to classify leaf images of plant diseases using the Plant Village dataset. Transfer learning using a pretrained SqueezeNet model was also tested for comparison purposes. The experimental results clearly demonstrate the generalization capabilities of the two architectures. Despite its perfect performance on the training data (100% accuracy), the Light CNN-29 model experienced a notable drop to 93% accuracy on both the validation and test sets. This discrepancy suggest that the Light CNN-29 model is overfitting, meaning the model has learned the training data so well that it has memorized specific features or noise. This hinders its ability to generalize new images. In contrast, SqueezeNet demonstrated superior generalization capabilities. The slight difference between its training accuracy (99%) and its validation and test accuracies (97%) suggests that the model learned the underlying patterns that distinguish leaf diseases rather than merely memorizing the training data. The high, balanced scores of precisions, recall, and F1 score (all 97% on the test set) show that SqueezeNet is accurate and robust making few false positive and false negative errors. Although the Light CNN-29 could not achieve the highest accuracy in the testing performance, its model still produced a satisfactory result with an accuracy above 90%.

# Acknowledgments

This work is supported by Bina Nusantara University as a part of Bina Nusantara University’s PIB-T entitled IoT-Based Vegetable Plant Pot Innovation In Residential Interiors as a Contribution To National Food Security with contract number: 85/VRRTT/V/2025 and contract date 02 Mei 2025. Several AI tools were used as assistants in the process of writing this paper. For example, the DeepL online tool was used for grammar checking and editing. The source code in this experiment can be accessed as open data at <http://doi.org/10.5281/zenodo.15967455>. For this study, Yulianto contributed to the writing and experimental details. Bambang Kartono Kurniawan contributed to the discussion of the research findings. Ida Bagus Ananta Wijaya contributed as the team leader, as well as to the literature review and discussion of the research findings. Gabriella Sagita Putri contributed to the conceptualization and discussion of the research findings. For Hamizah Abdul Hamid contributed to the methodology, editing, and discussion of the research findings.

# References

1. Chelladurai, A., Manoj Kumar, D. P., Askar, S. S., & Abouhawwash, M. (2025). Classification of tomato leaf disease using Transductive Long Short-Term Memory with an attention mechanism. *Frontiers in Plant Science*, *15*. https://doi.org/10.3389/fpls.2024.1467811
2. He, Y., Gao, Q., & Ma, Z. (2022). A Crop Leaf Disease Image Recognition Method Based on Bilinear Residual Networks. *Mathematical Problems in Engineering*, *2022*, 1–15. https://doi.org/10.1155/2022/2948506
3. Hughes, David. P., & Salathe, M. (2016). *An open access repository of images on plant health to enable the development of mobile disease diagnostics*.
4. Kathole, A., Shinde, S., & Wadhwa, L. (2024). Integrated MLOps and EEG techniques for enhanced crime detection and prevention. *Multidisciplinary Science Journal*, *6*(1). https://doi.org/10.31893/multiscience.2024009
5. Kumar Das, P. (2023). Leaf Disease Classification in Bell Pepper Plant using VGGNet. *Journal of Innovative Image Processing*, *5*(1), 36–46. https://doi.org/10.36548/jiip.2023.1.003
6. Li, J., Yin, Z., Li, D., Zhang, H., & Xu, M. (2025). An efficient non-parametric feature calibration method for few-shot plant disease classification. *Frontiers in Plant Science*, *16*. https://doi.org/10.3389/fpls.2025.1541982
7. Li, Y., & Lu, Y. (2023). Plant Disease Classification and Adversarial Attack based CL-CondenseNetV2 and WT-MI-FGSM. *International Journal of Advanced Computer Science and Applications*, *14*(4). https://doi.org/10.14569/IJACSA.2023.0140413
8. Li, Z., Li, C., Deng, L., Fan, Y., Xiao, X., Ma, H., Qin, J., & Zhu, L. (2022). Improved AlexNet with Inception-V4 for Plant Disease Diagnosis. *Computational Intelligence and Neuroscience*, *2022*, 1–12. https://doi.org/10.1155/2022/5862600
9. Liang, J., Liang, R., & Wang, D. (2025). A novel lightweight model for tea disease classification based on feature reuse and channel focus attention mechanism. *Engineering Science and Technology, an International Journal*, *61*, 101940. https://doi.org/10.1016/j.jestch.2024.101940
10. Lu, J., Shi, R., Tong, J., Cheng, W., Ma, X., & Liu, X. (2023). Lightweight Method for Plant Disease Identification Using Deep Learning. *Intelligent Automation & Soft Computing*, *37*(1), 525–544. https://doi.org/10.32604/iasc.2023.038287
11. Ouamane, A., Chouchane, A., Himeur, Y., Miniaoui, S., Atalla, S., Mansoor, W., & Al-Ahmad, H. (2025). Optimized Vision Transformers for Superior Plant Disease Detection. *IEEE Access*, *13*, 48552–48570. https://doi.org/10.1109/ACCESS.2025.3547416
12. Qiu, J., Lu, X., Wang, X., Chen, C., Chen, Y., & Yang, Y. (2024). Research on image recognition of tomato leaf diseases based on improved AlexNet model. *Heliyon*, *10*(13), e33555. https://doi.org/10.1016/j.heliyon.2024.e33555
13. Sadhasivam, M., Geetha, M. K., & Maria Britto, J. G. (2024). Deep learning based multi disease classification of plant leaves using light weight residual architecture. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(4), 4646. https://doi.org/10.11591/ijece.v14i4.pp4646-4654
14. Sadhasivam, M., Kalaiselvi Geetha, M., & Gladson Maria Britto, J. (2024). Efficient deep learning architecture for the classification of diseased plant leaves. *Indonesian Journal of Electrical Engineering and Computer Science*, *33*(1), 198. https://doi.org/10.11591/ijeecs.v33.i1.pp198-206
15. Sembiring, A., Away, Y., Arnia, F., & Muharar, R. (2023). Parallelization of Concise Convolutional Neural Networks for Plant Classification. *Journal of Ecological Engineering*, *24*(2), 61–71. https://doi.org/10.12911/22998993/156754
16. Tang, F., Porle, R. R., Tung Yew, H., & Wong, F. (2025). Identification of Maize Diseases Based on Dynamic Convolution and Tri-Attention Mechanism. *IEEE Access*, *13*, 6834–6844. https://doi.org/10.1109/ACCESS.2025.3525661
17. Verma, S., Chug, A., Singh, R. P., Singh, A. P., & Singh, D. (2021). SE-CapsNet: Automated evaluation of plant disease severity based on feature extraction through Squeeze and Excitation (SE) networks and Capsule networks. *Kuwait Journal of Science*, *49*(1). https://doi.org/10.48129/kjs.v49i1.10586
18. Wang, T., Xu, H., Hai, Y., Cui, Y., & Chen, Z. (2022). An Improved Crop Disease Identification Method Based on Lightweight Convolutional Neural Network. *Journal of Electrical and Computer Engineering*, *2022*, 1–16. https://doi.org/10.1155/2022/6342357
19. Wang, X., & Cao, W. (2023). Bit-Plane and Correlation Spatial Attention Modules for Plant Disease Classification. *IEEE Access*, *11*, 93852–93863. https://doi.org/10.1109/ACCESS.2023.3309925
20. Wu, X., He, R., Sun, Z., & Tan, T. (2018). A Light CNN for Deep Face Representation With Noisy Labels. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2884–2896. https://doi.org/10.1109/TIFS.2018.2833032
21. Yang, W., Yuan, Y., Zhang, D., Zheng, L., & Nie, F. (2024). An Effective Image Classification Method for Plant Diseases with Improved Channel Attention Mechanism aECAnet Based on Deep Learning. *Symmetry*, *16*(4), 451. https://doi.org/10.3390/sym16040451