

# Efficient Paprika Leaf Disease Classification Using MobileNetV2-Based Deep Learning and Image Augmentation

Abdul Harist<sup>1</sup>, Teguh Randi Angginta<sup>2</sup>, Siti Herawati<sup>3\*</sup>

<sup>1,2</sup> Program Studi Teknik Informatika, Jurusan Teknologi Informasi dan Komputer, Politeknik Negeri Lhokseumawe, Indonesia

<sup>3</sup> Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, Indonesia

\*Penulis Korespondensi: [2210631170153@student.unsika.ac.id](mailto:2210631170153@student.unsika.ac.id)

**Article info:** Received 10/12/2025, Revised 18/02/2026, Accepted 20/02/2026

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Abstract

The decline in productivity of paprika plants (*Capsicum annuum*) is often closely associated with leaf health disorders, particularly bacterial spot disease, which is difficult to identify at an early stage through conventional visual inspection. Early detection of this disease is crucial, as delayed diagnosis can lead to significant yield losses and reduced crop quality. However, manual monitoring relies heavily on expert knowledge and is prone to subjectivity and human error, especially in large-scale cultivation systems. To address this challenge, this study proposes a deep learning-based image classification approach for identifying diseases in paprika leaves by leveraging transfer learning with the MobileNetV2 architecture. MobileNetV2 was selected due to its lightweight structure and computational efficiency, making it suitable for practical and real-time agricultural applications. The dataset used in this research was obtained from the PlantVillage database and consists of two classes: healthy paprika leaves and leaves infected with bacterial spot disease. To enhance the robustness and generalization capability of the proposed model, the training data were enriched using various data augmentation techniques, including rotation, flipping, scaling, and brightness adjustment. These techniques help mitigate overfitting and improve the model's ability to recognize disease patterns under diverse imaging conditions. Experimental results demonstrate that the proposed model achieves stable and reliable classification performance, with an overall accuracy of 96%, accompanied by balanced precision and sensitivity values across both classes. These results indicate that MobileNetV2 is highly effective for paprika leaf disease classification. Furthermore, the findings suggest strong potential for implementing the proposed approach as an image-based plant disease detection system, supporting precision agriculture and enabling early intervention to improve crop productivity and sustainability.

**Keywords:** paprika, MobileNetV2, plant disease, image classification, transfer learning

## 1. Introduction

The agricultural sector is a major pillar of global food security. However, plant diseases remain a serious threat, causing significant losses in agricultural production. According to a report by the Food and Agriculture Organization (FAO), approximately 20-40% of annual crop yields are lost to diseases and pests, with a significant impact on horticultural commodities such as peppers, tomatoes, and potatoes, which are susceptible to disease.[1]. Peppers (*Capsicum annuum*) have high economic value, are widely cultivated, and are widely used in the culinary and industrial sectors. However, they are susceptible to Bacterial Spot caused by *Xanthomonas campestris* pv. *vesicatoria*. This disease causes brown spots that progress to necrosis, thus reducing the quality and quantity of production. Early detection of this disease is important so that farmers can take timely control measures.[2]. Plant disease diagnosis is carried out by plant pathologists through direct observation. However, this method is time-consuming, subjective, and impractical. Computer vision and deep learning technology can automatically identify diseases using leaf images.[3] The PlantVillage Dataset is a public dataset containing over 54,000 leaf images from 14 plant species across various disease categories and health conditions. This dataset provided images of bell pepper leaves in healthy and bacterial spot conditions for this study [4].

Several previous studies reported high accuracy using deep learning approaches. Khan et al. (2023) developed a lightweight CNN for edge computing with 95.6% accuracy on laboratory images, but performance decreased in field conditions with varying lighting [5]. Noyan (2022) found bias in PlantVillage due to images captured against a homogeneous background. The trained model achieved an internal accuracy of 99.2%, but dropped to 82.7% when tested on field images [6]. Pandey et al. (2024) compared CNNs, Random Forests, and SVMs. CNN achieved the highest accuracy (97.3%), but remained sensitive to lighting and background variations [7]. Other approaches have also been proposed to improve classification performance. Ouamane et al. (2024) proposed a CNN method based on Tensor Subspace Learning that achieves up to 99% accuracy, but requires high computational requirements for large-scale implementations. [8] Saleem et al. (2020) compared VGG16, ResNet50, InceptionV3, and Xception. Xception achieved 99.81% validation accuracy, but it still overfitted to the PlantVillage dataset. [9] These studies demonstrate that deep learning models achieve high accuracy, but the limited variation within PlantVillage poses generalization challenges. Data augmentation is an effective strategy for synthetically expanding the distribution of training images.

The objective of this study is to build and evaluate a Convolutional Neural Network (CNN)-based model for pepper leaf disease classification using a transfer learning approach with the MobileNetV2 architecture. Transfer learning aims to improve model performance, especially with limited training data, by leveraging pre-trained weights that have proven effective in image recognition tasks. The developed model focuses on classifying two main conditions of bell pepper leaves: healthy leaves and leaves infected with bacterial spot disease, based on digital images sourced from the PlantVillage dataset. Through the development of this model, the research is expected to produce an accurate, efficient, and reliable classification system for automatically detecting bell pepper leaf diseases. Furthermore, the proposed model aims to have low computational complexity, making it potentially applicable to devices with limited resources, such as mobile or embedded systems. Therefore, the results of this research are expected to contribute to the development of smart agriculture technology, particularly in supporting early detection of plant diseases to increase productivity and sustainability in the agricultural sector.

## 2. Method

The Convolutional Neural Network (CNN)-based paprika leaf disease classification model consists of two main stages: training and testing/classification.

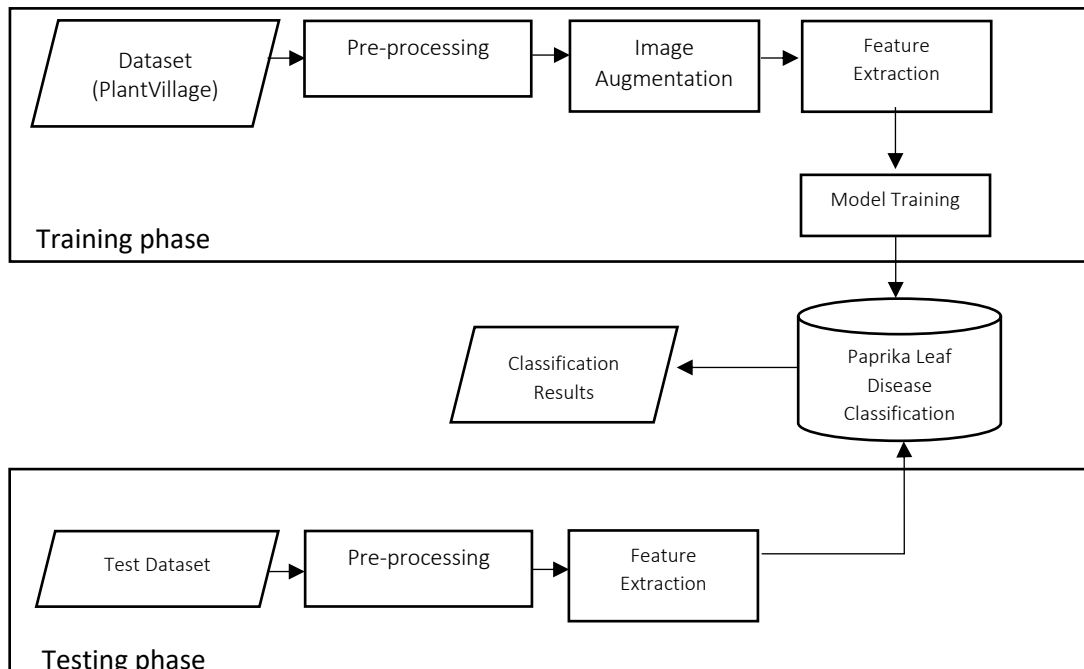




Figure 1. The Proposed Model Steps

During training, paprika leaf image data from the PlantVillage dataset undergoes preprocessing, including resizing, normalization, and data cleaning, to meet the model's requirements. Image augmentation is then performed to increase data variation, such as rotation, flipping, and scaling, to reduce the risk of overfitting. The augmented images are then processed using a CNN to extract important features from each image. These features are used during model training to build a paprika leaf disease classification model. During testing, the test images undergo preprocessing and feature extraction using the same CNN, then are fed into the trained model to produce paprika leaf disease classification results based on the detected classes. The detailed steps of the proposed model are shown in Figure 1.

## 2.1 Dataset

The dataset used in this study is the PlantVillage Dataset, which contains a collection of plant leaf images under controlled conditions with a uniform background to support plant disease detection research using computer vision techniques [11]. Of the 14 available plant classes, this study specifically uses a subset of bell pepper leaves, which consist of two classes: healthy (H) and bacterial spot (BS). All images in this dataset have a fixed resolution of  $256 \times 256$  pixels and are recorded in RGB format. In the healthy class, there are 1477 images, and in the bacterial spot class, there are 997 images, for a total of 2474 images used before the training process. These images display variations in leaf visual conditions, such as differences in color, texture, and the severity of disease symptoms, providing sufficient diversity for the model's learning process. Examples of leaf images from the dataset are shown in Table 1.

Tabel 1. Contoh gambar dari dataset PlantVillage

No.	Gambar	Class
		H
2		BS

## 2.2 Pre-processing

The preprocessing stage aims to improve image quality before it is processed by the CNN model during training. This process is important so that all images have uniform characteristics and formats [12]. In this study, preprocessing was applied to all paprika leaf images from the PlantVillage dataset before entering the augmentation and model training stages. The preprocessing stage includes adjusting image size and normalizing pixel values.

### 2.2.1 Resize

The first step in the preprocessing stage is image resizing. All images in the dataset initially have a resolution of  $256 \times 256$  pixels. To reduce computing load and speed up model training, the images are resized to  $128 \times 128$  pixels. This resizing process intends to balance computational effectiveness and the model's ability to capture relevant visual information. The resizing is applied consistently across all images in the dataset, guaranteeing uniform input dimensions and aiding the model's learning. With smaller yet representative image sizes, the model is expected to optimally extract key features, such as leaf patterns, textures, and shapes, without losing the image's main characteristics. An example of the results of resizing leaf images is shown in Figure 2.



(a) Pepper bell bacterial spot before resizing



(b) Pepper bell bacterial spot after resizing

Figure 2. Pepper bell bacterial spot before and after resizing

### 2.2.2 Rescale (Normalisasi Nilai Piksel)

After resizing, the next step in preprocessing is image normalization. Normalization converts the pixel intensity range from 0–255 to 0–1. This process is done by dividing each pixel value by a constant 255, so that all data is on a uniform scale. Normalization aims to stabilize the distribution of input data and reduce scale differences among features, which can affect the model's learning performance. With smaller, more controlled pixel values, the optimization process during training becomes more stable, enabling model convergence to be achieved more quickly. Although normalization does not visually change the image's appearance, this step plays a crucial role in improving the performance and accuracy of Convolutional Neural Network (CNN) models. An example of the resulting image after the normalization process is shown in Figure 2.



(c) Pepper bell bacterial spot sebelum normalisasi



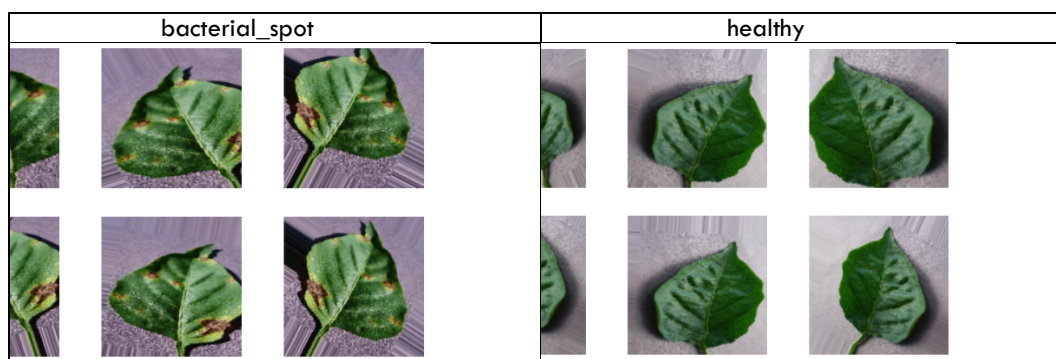
(d) Pepper bell bacterial spot setelah normalisasi

Gambar 1 pepper bell bacterial spot sebelum dan setelah normalisasi

### 2.2.3 Aumentasi Gambar

The data augmentation stage was conducted to increase the number and diversity of training images, thereby improving the model's generalization capabilities [13]. The augmentation process was applied to preprocessed images, namely  $128 \times 128$ -pixel normalized images. With augmentation, the model can recognize disease patterns on leaves under various visual conditions that may not be covered in the original dataset. Image augmentation was performed using ImageDataGenerator with several geometric transformations. Image rotation was applied with a maximum angle range of 20 degrees to simulate variations in leaf orientation. In addition, horizontal and vertical image shifts were applied to 10% of the original size in each direction to represent differences in object positions in the image. Zooming techniques were also applied, with a factor of up to 20%, to account for variations in leaf scale. A horizontal flip was also used to increase the variation in leaf viewpoints, while a vertical flip was not applied. The entire augmentation process was performed only on the training data, while the validation data was not augmented and was only normalized. With this configuration, the model was trained on more diverse images while preserving the original visual characteristics, thereby increasing its resilience to image variations and reducing overfitting. The image results are shown in Table 2.

Tabel 2. Augmentation Results



## 2.3 Convolutional Neural Network (CNN)

After completing the preprocessing and data augmentation stages, the next step is to apply a Convolutional Neural Network (CNN) to classify paprika leaves into healthy and bacterial spot–infected categories. The CNN model is trained on labeled images from the PlantVillage dataset to learn discriminative visual features, including texture, color patterns, and lesion characteristics. Once the training phase is complete, the trained model is evaluated on test data to assess its ability to accurately classify unseen paprika leaf images and to measure its overall performance.

### 2.3.1 Transfer learning

A transfer learning approach was used to classify pepper leaf diseases. Transfer learning enables the model to leverage weights trained on large datasets, such as ImageNet, resulting in faster training and more stable performance despite relatively limited datasets [14]. The underlying architecture is MobileNetV2, a lightweight CNN model designed for resource-constrained environments. MobileNetV2 has proven efficient in various image classification tasks [15]. In this study, all base layers of MobileNetV2 are frozen, serving as feature extractors without retraining.

### 2.3.2 CNN Model Architecture

In this study, the model was developed using MobileNetV2 with `include_top=False`, eliminating the model's built-in classification and using only its feature extraction. The output from MobileNetV2 was then connected to several additional layers to classify two classes (healthy and bacterial spots).

The additional architectures used include:

- GlobalAveragePooling2D, converting the output feature map into a 1-dimensional vector.
- Dropout (rate = 0.5), to reduce overfitting.
- Dense (128 neurons, ReLU activation), a fully connected layer for learning high-level representations.
- Dense (1 neuron, Sigmoid activation), because the output is only a single probability value (0 = healthy, 1 = bacterial).

MobileNetV2 acts as an automatic feature extractor, while the additional layers perform the classification specifically for the bell pepper dataset.

### 2.3.3 Kompilasi Model

The model was compiled using:

Loss function: Binary Cross-Entropy (BCE), an output layer that produces a single probability value, where values close to 0 indicate the healthy class and values close to 1 indicate the bacterial spot class.

$$\text{BCE}(y, y^{\wedge}) = -[y \cdot \log(y^{\wedge}) + (1 - y) \cdot \log(1 - y^{\wedge})]^{1} \quad (1)$$

Optimizer: Adam, with a learning rate of 0.0001 for learning stability.

Metrics: accuracy, to measure the model's ability to correctly classify images.

### 2.3.4 Proses Pelatihan

The model is trained using augmented data in the training phase, while the validation data is simply rescaled without augmentation.

Training parameters used:

- Epoch: 20, the model will view the entire dataset 20 times.
- Batch size: 40, the model processes 40 images simultaneously.

The training data uses images that have undergone preprocessing and augmentation. The testing data uses preprocessed images (without augmentation).

Training is performed using the `model.fit()` method, with input provided as a generator (e.g., `ImageDataGenerator`) or `tf.data`. A dataset containing a collection of images that have undergone preprocessing and augmentation. This process allows the model to receive batches of images incrementally, reducing memory consumption.

## 3. Results and Discussion

The MobileNetV2 model was trained by monitoring changes in accuracy and loss at each epoch. The data visualization in Figure 3 shows that in the early epochs (1-3), training accuracy increased significantly from around 70% to nearly 90%, suggesting that the MobileNetV2 model with transfer learning quickly captured the basic pattern of bell pepper leaf images. In this phase, validation accuracy also immediately reached above 90%, indicating that the initial features learned were already relevant to the validation data. In the middle epochs (4-10), the increase in training and validation accuracy was more gradual and stable, with values ranging from 94-95% each without significant differences, indicating a balanced learning process and no overfitting. Entering the final training phase (11-20), both accuracy curves tended to stagnate around 95-96%, indicating the model had reached convergence and optimal performance on the used dataset, with small fluctuations within reasonable limits.

In terms of loss, the graph in Figure 3 shows a consistent downward trend across both the training and validation data. In the early training phase (epochs 1-3), the training loss dropped sharply from around 0.60 to below 0.30, while the validation loss dropped from around 0.35 to around 0.20. This rapid decline indicates that the model effectively reduces prediction errors in the early stages of learning. In the later phase (epochs 4-20), the loss decline is more stable, with the training loss in the range of 0.15-0.18 and the validation loss in the range of 0.12-0.15. There is no significant increase in validation loss or divergence between the two curves, a strong indicator that the model successfully avoids overfitting and has optimal generalization capabilities to the validation data.

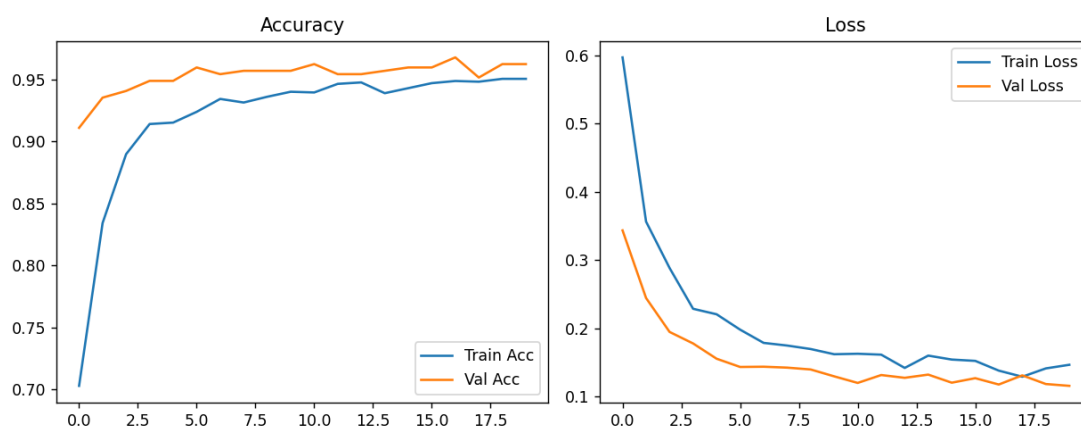


Figure 3. Accuracy and loss of model training

After the training phase, a performance evaluation was conducted using a confusion matrix to assess the model's ability to classify paprika leaf images into two classes: healthy and bacterial spots. The visualization in Figure 4 shows that, out of a total of 374 test data sets, the model correctly classified 148 healthy leaf images (true negatives) and 211 leaf images infected with bacterial spots (true positives).

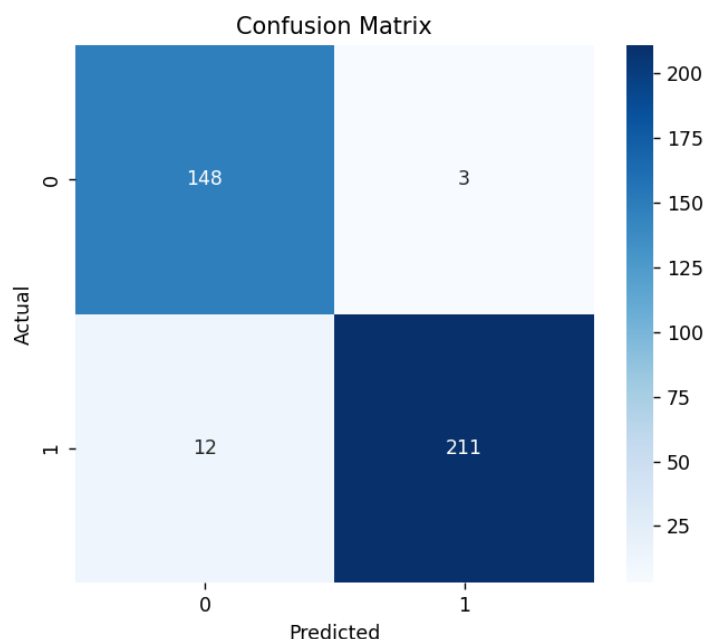


Figure 4. Model's Confusion matrix on Testing phase

On the other hand, three healthy leaf images were misclassified as infected (false positives), and 12 infected leaf images were misclassified as healthy (false negatives). This relatively small number of misclassifications indicates that the model has strong class separation and does not exhibit significant bias toward any one class. Overall, the resulting prediction distribution confirms the previous training results, showing that the MobileNetV2 model achieved high accuracy in classifying paprika leaf images. The classification report in Table 3 confirms these results. For the bacterial\_spot class, the model achieved a precision of 0.93 and a recall of 0.98, indicating excellent detection of bacterially infected leaves with a low error rate. For the healthy class, a precision of 0.99 and a recall of 0.95 indicate that the model recognized healthy leaves with high accuracy. The consistent F1-score values across both classes, 0.95 and 0.97, demonstrate a balance between precision and recall.

Overall, the model achieved an accuracy of 0.96, indicating strong classification performance on the paprika leaf dataset.

Tabel 3 Laporan Kinerja Klasifikasi Model CNN pada Data Uji.

	Precision	Recall	F1-score	support
Bacterial_spot	0.93	0.98	0.95	151
Healthy	0.99	0.95	0.97	223
Accuracy			0.96	374
Macro avg	0.96	0.96	0.96	374
Weighted avg	0.96	0.96	0.96	374

#### 4. Conlusions

This study successfully developed a model for paprika leaf disease classification using a lightweight, efficient MobileNetV2-based transfer learning approach. Evaluation results showed the model achieved 96% accuracy, demonstrating excellent ability to distinguish healthy paprika leaves from those infected with bacterial spot. The consistent pattern of accuracy improvement and reduction in loss across training and validation data indicates that the model does not overfit and has good generalization capabilities. Thus, MobileNetV2 has proven effective for

paprika leaf disease classification and has the potential for further development through additional dataset variations and hyperparameter optimization.

## Reference

- [1]. V. K. Singh, R. Singh, A. Kumar, and R. Bhadouria, "Current status of plant diseases and food security," *Food Security and Plant Disease Management*, pp. 19–35, Jan. 2021, doi: 10.1016/B978-0-12-821843-3.00019-2.
- [2]. M. Dai *et al.*, "Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks," *Front Plant Sci*, vol. 14, 2023, doi: 10.3389/fpls.2023.1230886.
- [3]. B. Mishra and B. Chourasia, "Deep Learning and Image Processing Techniques for Plant Disease Detection: A Review," *International Journal of Recent Development in Engineering and Technology Website: www.ijrdet.com*, vol. 14, no. 10, p. 1, 2025, [Online]. Available: [www.ijrdet.com](http://www.ijrdet.com)
- [4]. M. P. Mathew, S. Elayidom, J. V Raj P, and A. K. M, "Enhanced Bell Pepper and Grape Leaf Disease Classification Using a Depthwise Separable VGG19-Capsule Network," *Electronic Letters on Computer Vision and Image Analysis*, vol. 24, no. 1, pp. 179–194, 2024.
- [5]. A. T. Khan, S. M. Jensen, A. R. Khan, and S. Li, "Plant disease detection model for edge computing devices," *Front Plant Sci*, vol. 14, 2023, doi: 10.3389/fpls.2023.1308528.
- [6]. M. A. Noyan, "Uncovering bias in the PlantVillage dataset," *ArXiv*, Jun. 2022, Accessed: Dec. 23, 2025. [Online]. Available: <http://arxiv.org/abs/2206.04374>
- [7]. V. Pandey, U. Tripathi, V. K. Singh, Y. S. Gaur, and D. Gupta, "Survey of Accuracy Prediction on the PlantVillage Dataset using different ML techniques," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024, doi: 10.4108/eetiot.4578.
- [8]. A. Ouamane *et al.*, "Enhancing Plant Disease Detection: A Novel CNN-Based Approach with Tensor Subspace Learning and HOWSVD-MD," *Neural Comput Appl*, vol. 36, no. 36, pp. 22957–22981, May 2024, doi: 10.1007/s00521-024-10454-1.
- [9]. M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers," *Plants*, vol. 9, no. 10, pp. 1–17, Oct. 2020, doi: 10.3390/plants9101319.
- [10]. T. Ilyas, H. Jin, M. I. Siddique, S. J. Lee, H. Kim, and L. Chua, "DIANA: A deep learning-based paprika plant disease and pest phenotyping system with disease severity analysis," *Front Plant Sci*, vol. 13, Oct. 2022, doi: 10.3389/fpls.2022.983625.
- [11]. Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecol Inform*, vol. 61, p. 101182, Mar. 2021, doi: 10.1016/J.ECOINF.2020.101182.
- [12]. Heru *et al.*, "Computer Vision: Classification of Images Based On Deep Learning with the CNN Architecture Model," *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, vol. 9, no. 11, pp. 2394–2320, 2022.
- [13]. M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Elsevier*, vol. 137, Nov. 2022, doi: 10.1016/j.patcog.2023.109347.
- [14]. I. Haque, M. A. Islam, K. Roy, M. M. Rahaman, A. A. Shohan, and M. S. Islam, "Classifying Pepper Disease based on Transfer Learning: A Deep Learning Approach," *Proceedings - International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2022*, pp. 620–629, 2022, doi: 10.1109/ICAAIC53929.2022.9793178.
- [15]. P. Xiao, Y. Pang, H. Feng, and Y. Hao, "Optimized MobileNetV2 Based on Model Pruning for Image Classification," *2022 IEEE International Conference on Visual Communications and Image Processing, VCIP 2022*, 2022, doi: 10.1109/VCIP56404.2022.10008829.